

Institut für Telematik
Cooperation & Management
Prof. Dr. Sebastian Abeck



Rollenmodelle für die Zugriffskontrolle in Unternehmen

Diplomarbeit
von

Korbinian Molitorisz

Verantwortlicher Betreuer:	Prof. Dr. Sebastian Abeck
Betreuender Mitarbeiter:	Dr.-Ing. Christian Emig

Bearbeitungszeit: 01. Juni 2008 – 21. November 2008

Universität Karlsruhe (TH) · Institut für Telematik · Geb. 20.20 · Zirkel 2 · 76131 Karlsruhe

Ehrenwörtliche Erklärung

Ich erkläre hiermit, die vorliegende Arbeit selbstständig verfasst und keine anderen als die angegebenen Quellen und Hilfsmittel verwendet zu haben.

Karlsruhe, den 21. November 2008

Korbinian Molitorisz

Inhaltsverzeichnis

1 EINLEITUNG

In den letzten Jahren wurde im Bereich der Zugriffskontrolle der Begriff „Rolle“ sowohl in der Forschung, aber auch in der Industrie sehr stark diskutiert und untersucht. Während die Forschung primär daran interessiert war, theoretische Modelle zu entwickeln, orientierte sich die Industrie sehr stark an den konkreten Bedürfnissen von Unternehmen [Ka07b]. Diese versprachen sich von einer rollenbasierten Zugriffskontrolle angesichts immer komplexerer Infrastrukturen zeitliche Einsparungen und effizienteres Arbeiten in Verwaltungstätigkeiten, aber auch eine bessere Einhaltung gesetzlicher Vorgaben. Die Zielsetzung der Untersuchungen war somit sehr unterschiedlich, was dazu führte, dass zwischen den beiden Herangehensweisen an diese Thematik eine große Lücke entstand.

In dieser Arbeit werden Rollenmodelle für die rollenbasierte Zugriffskontrolle entwickelt, die sich besser an die Gegebenheiten in Unternehmen anpassen und so die Lücke zwischen theoretischen Rollenmodellen und praktischen Umsetzungen von RBAC geschlossen werden kann. Eine zentrale Forderung hierbei ist die explizite Trennung unterschiedlicher Rollentypen. Dies kommt sowohl in der Modellspezifikation, als auch in der funktionalen Spezifikation des Rollenmodells zum Tragen. Zusätzlich zur Entwicklung des Rollenmodells wird ein Vorgehensmodell entwickelt, das zur technischen Umsetzung des Rollenmodells herangezogen wird. Die zentralen Aspekte des Vorgehens sind dabei die Unterstützung der Entwicklung unterschiedlicher Rollentypen einerseits und die explizite Modellierung einer Phase für den produktiven Einsatz des instanziierten Rollenmodells andererseits. Zum Tragfähigkeitsnachweis wird ein Beispielszenario vorgestellt, in dem das Rollenmodell gemäß der Vorgaben des Vorgehensmodells in einem Werkzeug für den Bereich des Rollenmanagements realisiert wird. Diese Arbeit entstand in Kooperation mit der Firma iC Consult, Gesellschaft für Systemintegration und Kommunikation mbH, die sich an dieser Arbeit durch ihre Erfahrungswerte aus der Praxis aktiv beteiligt hat.

1.1 Einführung in das Themengebiet

Der Bedarf an Mechanismen zur Wahrung der Sicherheit besteht, seitdem es zu schützende Informationen gibt [FK+07]. Der Ruf nach Prinzipien zur Sicherheit von Computersystemen entwickelte sich im Einklang mit dem Aufkommen von *Time-sharing* Systemen ab dem Jahr 1960, die es erstmals erlaubten, mehrere Benutzer über Terminal-Verbindungen gleichzeitig auf einem Rechner arbeiten zu lassen [Ta02]. Durch das starke Engagement des amerikanischen Verteidigungsministeriums etablierte sich der Schutzbedarf von Systemen zu einem wichtigen Aspekt und wurde mit dem Aufkommen vernetzter Systeme und des Internet zu einem inhärenten Entwurfsziel in der IT-Landschaft [FK+07]. Dieser Sicherheitsbegriff ist sehr facettenreich und wird übereinstimmend in die Klassen „Funktionssicherheit“, „Informationssicherheit“ sowie „Datensicherheit“ unterteilt. Ein System gilt als funktionssicher, wenn die Ist-Funktionalität mit der Soll-Funktionalität übereinstimmt. Die Sicherheit von Informationen und Daten setzt ein funktionssicheres System voraus und gewährleistet, dass nur in autorisierter Weise auf Informationen und Daten zugegriffen werden kann. Eine wesentliche Folgerung der Sicherheit von Informationen und Daten ist, dass es sich bei der Aufrechterhaltung eines bestimmten Sicherheitsniveaus um einen fortlaufenden Prozess handelt, da sich die Eigenschaften von Systemen im Laufe der Zeit ändern [Ec05]. Adäquate Verwaltungsmechanismen und die Kontrolle von Zugriffsberechtigungen sind zur Erhaltung dieses Sicherheitsniveaus fundamental [WW07]. Zugriffskontrollarchitekturen stellen somit einen wesentlichen Baustein heutiger Computersysteme dar.

Zur Umsetzung der Zugriffskontrolle stellte [La69] im Jahr 1969 ein formales Beschreibungsmodell vor. In diesem Rahmen wurden die Konzepte Subjekt (engl *subject*) und Objekt (engl. *object*) eingeführt und die Zugriffskontrolle über eine Zugriffsmatrix (engl. *access matrix*) vorgenommen, die Subjekt und Objekt direkt miteinander verknüpft. Dieses Modell war sehr abstrakt gehalten und wurde in den Jahren ab 1970 mathematisch spezifiziert, worauf sich ein all-

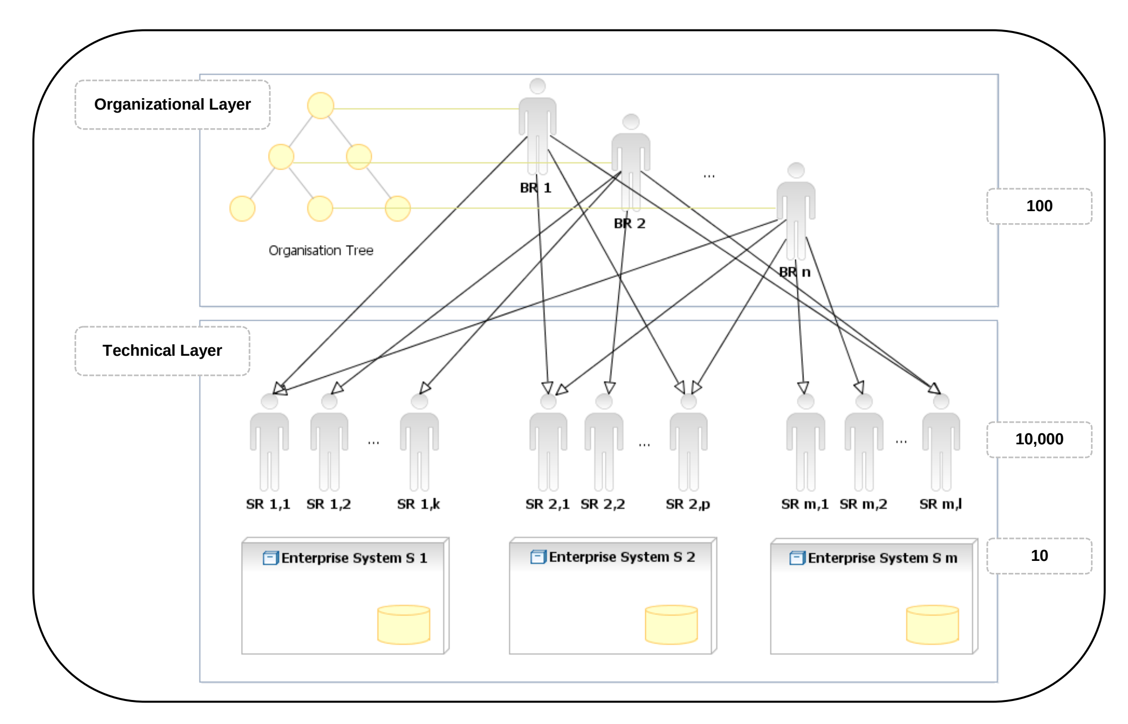
gemeines Verständnis einer aktiven Einheit (*subject*) und einer passiven Einheit (*object*) zur Beschreibung von Zugriffsversuchen herausbildete. Die Zugriffsmatrix ist in Form einer Relation zwischen Subjekt und Objekt realisiert, die die Zugriffsrechte spezifiziert und so den Berechtigungsgrad definiert. Technisch wird dieser Sachverhalt oft in Form von Zugriffskontrollliste (engl. *access control list*, ACL) realisiert, über die jede Ressource selbst verfügt. Dieses Modell ist heute bekannt als identitätsbasierte Zugriffskontrolle (engl. *identity-based access control*, IBAC) [C&M-A-ID], weil die Zugriffsrechte direkt für die Identität des zugreifenden Subjekts definiert sind [EB+07]. Zwar ist dieser Architekturstil leicht verständlich und durch die Relation zwischen Subjekt und Objekt leicht implementierbar, jedoch weist er zwei wesentliche Nachteile gerade in vernetzten Unternehmenslandschaften auf: Erstens skaliert IBAC für steigende Benutzerzahlen und Ressourcen sehr schlecht, was daran liegt, dass jede Ressource über eine eigene Zugriffskontrollliste verfügt und zweitens stehen die Identitäten der Benutzer nicht in Beziehung zu deren geschäftlichen Funktion. Wenn ein Benutzer innerhalb eines Unternehmens die Position wechselt und somit andere Aufgaben erfüllt, müssen die Zugriffskontrolllisten aller Systeme einzeln geändert werden, was sehr ineffizient ist [C&M-A-ID]. Zusätzlich ist hierbei nicht gewährleistet, dass ein Benutzer nur über das Maß an Rechten verfügt, das er zur Durchführung seiner Aufgaben benötigt, was zu Inkonsistenzen zwischen den erteilten und den benötigten Rechten führt. In der Realität werden Zugriffsrechte somit oftmals inkorrekt zugeteilt und die Gesamtstruktur weist Inkonsistenzen auf.

In den Jahren um 1990 etablierte sich ein verbessertes Modell zur Zugriffskontrolle. Hierbei wurde das Konzept der „Rolle“ als Indirektionsstufe zwischen *subject* und *object* eingeführt. Dieser Architekturstil, der die Skalierungsprobleme von IBAC beheben will, ist die rollenbasierte Zugriffskontrolle (engl. *role-based access control*, RBAC) [C&M-A-ID]. Anders als beim IBAC-Modell, in dem die Zugriffskontrolle direkt zwischen Identitäten und Berechtigungen umgesetzt ist, spiegeln die Zugriffskontrolllisten bei RBAC die Relation zwischen Rollen und Berechtigungen wider. Die Beziehung zwischen Identität und Rolle wird durch eine zusätzliche Relation dargestellt. Semantisch gesehen entspricht eine Rolle somit einer Geschäftsfunktion, die über eine Menge an Berechtigungen in verschiedenen Systemen verfügt. Das Zusammenfassen von Berechtigungen zu Rollen erleichtert zum Einen das Hinzufügen oder Entfernen von Berechtigungen an Identitäten und zum Anderen das Sicherstellen, dass ein angemessener Berechtigungsumfang zugewiesen wurde [C&M-A-ID]. Forschungseinrichtungen erkannten diesen Mehrwert von Rollen als Abstraktionseinheit für Zugriffsrechte in Computersystemen gegenüber der direkten Kopplung von Berechtigungen und Benutzern [FK+07]. In den letzten Jahren entstand daher eine lebhafte Diskussion innerhalb der Forschung mit dem Ziel der Formulierung theoretischer Standardrahmenwerke für die rollenbasierte Zugriffskontrolle [Ka07a]. Im Bereich der Technik lässt sich heute feststellen, dass Rollenmodelle mit einer differenzierten Betrachtung des Rollenbegriffs nötig sind, um die unterschiedlichen Rollentypen, die faktisch in Unternehmen vorhanden sind, bereits auf Modellebene darstellbar zu machen. So entstanden in jüngster Zeit mehrere Rollenmodelle für die rollenbasierte Zugriffskontrolle mit unterschiedlichen Zielsetzungen: Das NIST-RBAC-Standardrahmenwerk definiert vier Rollenmodelle mit unterschiedlicher funktionaler Spezifikation, lässt dabei jedoch eine differenzierte Betrachtung des Rollenbegriffs vermissen. Das ERBAC-Rollenmodell hingegen, welches durch praktische Erfahrungen im Umgang mit Rollen inspiriert wurde, überträgt den Rollenbegriff auf die technische Umsetzung in Unternehmen und interpretiert eine Rolle als systemübergreifende Einheit zur Kapselung von Berechtigungen. Auch hier findet sich keine explizite Unterscheidung des Rollenbegriffs vor. Die bisherigen Rollenmodelle befassen sich lediglich mit der reinen Konzeption und funktionalen Spezifikation, nicht jedoch mit der Entwicklung eines für dieses Modell angemessene Vorgehensmodell zur Instanziierung und Überführung in den Wirkbetrieb. Die Aktualität dieses Themas, die breite Akzeptanz von RBAC innerhalb von Forschung und Industrie sowie der bisher sehr wenig beachtete Zusammenhang von Rollen- und Vorgehensmodellen stellen die Motivation für diese Forschungsarbeit dar.

1.2 In der Arbeit behandelte Fragestellungen

In den letzten Jahren ist ein deutlicher Zuwachs an rollenbasierten Zugriffskontrollarchitekturen zu beobachten, welche ein Architekturprinzip verkörpern, das sich der wachsenden Heterogenität in IT-Landschaften stellt und Lösungswege für die damit einhergehende Komplexität anbietet [Ka07a]. Aus diesem Grund sind rollenbasierte Zugriffskontrollarchitekturen sehr attraktiv für große Einrichtungen wie beispielsweise Unternehmen geworden. Obwohl RBAC ein vielversprechender Ansatz für Unternehmen darstellt, ist die technische Realisierung von RBAC im Unternehmenskontext allerdings nicht trivial. Dies wird klar, wenn man sich verdeutlicht, dass es heutzutage mehrere unterschiedliche Auffassungen des Rollenbegriffs gibt. In der Literatur existieren Rollen als Einheit für technische Berechtigungen ebenso, wie als Einheit für die Beschreibung von Aufgaben innerhalb einer Organisation. An dieser Stelle ist oftmals die Rede von Unternehmensrollen (engl *enterprise roles*) oder Geschäftsrollen (engl. *business roles*), während bei technischen Berechtigungen von IT-Rollen (engl. *IT roles*) oder technischen Rollen (engl. *technical roles*) gesprochen wird. Um diesen Sachverhalt zu vereinheitlichen, wird in dieser Arbeit durchgängig von „Geschäftsrollen" und „Systemrollen" gesprochen. Die vorliegende Arbeit knüpft mit der Forderung einer expliziten Trennung des Rollenbegriffs auf Modellebene genau an dieser Stelle an und entwickelt ein Rollenmodell, welches diesen Unterschied explizit formalisiert. Dieses Ziel orientiert sich daran, dass es prinzipiell zwei unterschiedliche Sichten auf ein Unternehmen gibt, was sich in diesen beiden Auffassungen des Rollenbegriffs ausdrückt: Die eine Sicht ist sehr technisch orientiert, betrachtet die eingesetzten Systeme sowie die darin enthaltenen Zugriffsrechte. Hier stellt eine Rolle eine Menge von Zugriffsrechten auf ein oder mehrere Systeme dar, was ohne Beschränkung der Allgemeinheit als Systemrolle bezeichnet wird. Im Gegensatz dazu abstrahiert die zweite Sicht von technischen Details und betrachtet die Organisationsstrukturen innerhalb des Unternehmens und dessen Geschäftsfunktionen, was sich etwa in Geschäftsprozessen zeigt. Der Begriff „Rolle" steht hier als Ausdruck für die handelnde bzw. aktive Einheit innerhalb von Geschäftsprozessen. Um diesen Unterschied klar zu machen, wird im Folgenden von Systemrollen (engl. *system roles*, SR) und Geschäftsrollen (engl. *business roles*, BR) gesprochen.

Die steigende Komplexität zeigt sich auf diesen Ebenen in unterschiedlicher Weise: Auf der technischen Ebene stellt man fest, dass die IT-Landschaft in Unternehmen aus einer Vielzahl unterschiedlicher Systeme besteht und demnach sehr heterogen ist [WW07]. Einerseits befinden sich Systeme unterschiedlicher Technologien und Hersteller im Einsatz und andererseits gibt es pro System eine Vielzahl unterschiedlicher Berechtigungsgrade. Dies führt zu einer großen Zahl an Systemrollen, die manuell nicht mehr effektiv verwaltet werden kann und den Ruf nach klaren Strukturen lauter werden lässt. Aus Unternehmenssicht stellt sich die Komplexität anders dar: Hier hat man es mit gewachsenen Unternehmensstrukturen, mit mehreren Abteilungen und Geschäftsfunktionen zu tun. Zwar können die Abteilungen, etwa in Form eines Organisationsdiagramms, in eine klare Struktur gebracht werden, jedoch sind die Geschäftsfunktionen in der Praxis oft nicht in logischer oder hierarchischer Weise, sondern beliebig über das ganze Unternehmen verteilt. Demgegenüber stehen gesetzliche Regularien wie beispielsweise der *Sarbanes-Oxley Act* (SOX) [Sen02] oder das Gesetz zur Kontrolle und Transparenz im Unternehmensbereich (KonTraG) [Bun98], die unter Anderem auch eine Erhöhung der Transparenz bei Strukturen, Befugnissen und Kompetenzen fordern. Dies erhöht auch hier den Anreiz nach einer klaren Organisation. Insgesamt betrachtet nimmt also die Komplexität sowohl auf Geschäftsebene, als auch auf technischer Ebene sehr stark zu, so dass ein Ansatzpunkt für diese Arbeit die Komplexität darstellt, der man sich heute durch den intensiven Gebrauch von Computersystemen und damit einhergehenden gesetzlichen Regularien ausgesetzt sieht. Es wird versucht, die Reduktion von Komplexität einerseits durch die explizite Trennung geschäftlicher- und technischer Aspekte im Rollenmodell zu ermöglichen sowie andererseits durch Mechanismen, die durch die explizite Trennung erst möglich werden. Information 1 veranschaulicht die Komplexität auf diesen beiden Ebenen und stützt sich dabei auf Erfahrungswerte der Firma iC Consult.



Information 1: Introduction – Clarification of the Problem

Information 1 zeigt den Sachverhalt, wie er sich heute in einem mittelständischen- aber auch einem Großunternehmen darstellt. Die Erfahrung aus Industrieprojekten zeigt, dass die Anzahl der eingesetzten Systeme S_i in der Größenordnung von etwa 10 liegt. Jedes System S_i verfügt über *k* eigene Systemrollen $SR_{i,1}$, $SR_{i,2}$, …, $SR_{i,k}$, was in der Summe aller Systemrollen der Größenordnung 10.000 entspricht. Eine Geschäftsrolle BR_j, von denen meist um die 100 verschiedenen Rollen existieren, akkumuliert eine Teilmenge dieser Systemrollen [Ka07c]. Der entscheidende Punkt ist nun, dass das Zuordnen von Geschäfts- und Systemrollen in Anbetracht der Zahlen manuell nicht mehr in korrekter und konsistenter Weise erledigt werden kann. Dies verdeutlicht die Notwendigkeit nach Maßnahmen, die diese Zuordnung unterstützen, wie etwa Rollenmanagementwerkzeuge. Diese stellen in Aussicht, den Aufbau einer rollenbasierten Gesamtstruktur zu unterstützen. Leider gibt es bisher kein standardisiertes Vorgehen, um unternehmensweit Geschäfts- und Systemrollen zu identifizieren, weil das hierfür benötigte Wissen über verschiedene Abteilungen verteilt ist. Zum Einen kann das technische Personal entscheiden, über welche Rechte ein Benutzer verfügen muss, da es die technische Seite des oder der Systeme kennt. Die Sichtweise dieses Personals ist oftmals sogar beschränkt auf nur ein einzelnes System, das es selbst verantwortet. Zum Anderen hat die Personalabteilung den Überblick über die Geschäftsfunktion dieses Benutzers, jedoch fehlt hier der technische Einblick in die vorhandenen Systeme. Somit erstreckt sich der Prozess sowohl über eine Verantwortlichkeitskette und damit einhergehend auch über einen langen Zeitraum.

Der zweite Schwerpunkt dieser Arbeit liegt darin, ein Vorgehensmodell zu entwickeln, das die Verknüpfung von Geschäfts- und Systemrollen im Unternehmenskontext unterstützt sowie die Entwicklung von Rollen als ganzheitliche Aufgabe im Sinne eines Lebenszyklus modelliert. Es basiert auf dem entwickelten Rollenmodell, unterscheidet bei der Entwicklung von Geschäfts- und Systemrollen zwischen zwei unterschiedlichen Vorgehensweisen und ermöglicht somit eine strukturierte und klare Entwicklung dieser beider Rollentypen. Bei der Identifikation von Rollen in einer gewachsenen Unternehmensstruktur existieren drei unterschiedliche Vorgehensweisen, die als Grundlage für die Instanziierung des Rollenmodells verwendet werden: Beim *top-down*-Vorgehen wird von den Unternehmensstrukturen und Geschäftsfunktionen ausgegangen und versucht, diese Funktionen in wohldefinierte Geschäftsrollen darzustellen und schrittweise zu verfeinern. Beim *bottom-up*-Vorgehen versucht man hingegen, Berechtigungen – ausgehend von technischen Zugriffsrechten – schrittweise zusammenzufassen und somit allgemeine Sys-

temrollen zu konzipieren. Die Frage, die sich hier stellt ist, in welchen Situationen diese Vorgehen Vorteile aufweisen und wie sie sich entsprechend kombinieren lassen. Das dritte Vorgehen (*middle-out*-Vorgehen) steht für einen hybriden Ansatz, der beide Vorgehensweisen kombiniert [SA+04]. Das Ziel des Vorgehensmodells ist, dass es sich bei der Identifikation von Rollen besser an die Gegebenheiten von Unternehmen anpasst als bisherige Modelle, sich dabei auf den hybriden Ansatz stützt und so die Lücke zwischen Theorie und Praxis schließt.

Der dritte Schwerpunkt dieser Arbeit liegt in einer kriteriengestützten Untersuchung ausgewählter Werkzeuge, die die Arbeitsprozesse zum Verwalten und Verknüpfen von Rollen unterstützen. Speziell die Aufgabe der Herausarbeitung von Rollen aus einer bestehenden Unternehmensstruktur (bekannt als *role discovery* oder *role mining*) gilt aktuell als sehr schwierig umzusetzen. Da die Berechtigungen auf technischer und organisatorischer Ebene im Zuge der Komplexität in Unternehmensstrukturen weder konsistent noch korrekt sind, fehlt eine ordentliche Datenbasis für das automatisierte Erzeugen von Rollen. Daran schließt sich das Verknüpfen dieser Rollen (bekannt als *role reconciliation*) an [Ka07b]. Das erste Kriterium zur Auswahl der in dieser Arbeit verwendeten Werkzeuge ist, dass sie ein hybrides Vorgehensmodell unterstützen und nicht für einen spezifischen Einsatzzweck hin konzipiert wurden. Das zweite Kriterium ist die technologische Reife der Werkzeuge, was daran gemessen wird, wie lange sie schon auf dem Markt angeboten werden. Dabei soll in dieser Arbeit ein bereits seit längerer Zeit verfügbares und somit etabliertes Werkzeug und ein relativ junges Werkzeug verwendet werden, um die unterschiedlichen Architekturen aufzuzeigen. Die behandelten Werkzeuge sind der Role Manager der Firma Sun sowie der Identity Manager der Firma Omada. Beide entstammen dem Marktsegment der Universalwerkzeuge, die sich mit allen Prozessen beschäftigen, die im Lebenszyklus einer Rolle anfallen. Alle Lösungen in diesem Marktsegment verbindet, dass sie Rollen systemübergreifend definieren und verwalten können und auf keinen speziellen Einsatzzweck hin entwickelt wurden. Darüber hinaus können sie Rolleninformationen an Anwendungen weiterreichen, die ihrerseits zwar mit Rollen als Einheit, nicht aber mit systemübergreifenden Rollen operieren [Ka07b]. Um eine Vergleichbarkeit dieser beider Werkzeuge zu ermöglichen, stellt diese Arbeit ein Szenario vor und bewertet anhand eines in dieser Arbeit entwickelten Kriterienkatalogs, wie gut sie in diesem Szenario das Ausarbeiten anhand des Vorgehensmodells unterstützen. Das dritte Ziel ist demnach eine Untersuchung der ausgewählten Werkzeuge bei der initialen Umstellung eines Unternehmens auf eine rollenbasierte Struktur. Es stellt sich die Frage nach der Qualität der Unterstützung bei dieser speziellen Aufgabe innerhalb des gesamten Bereichs des Rollenmanagements.

Zusammengefasst beschäftigt sich diese Arbeit mit folgenden Aspekten:

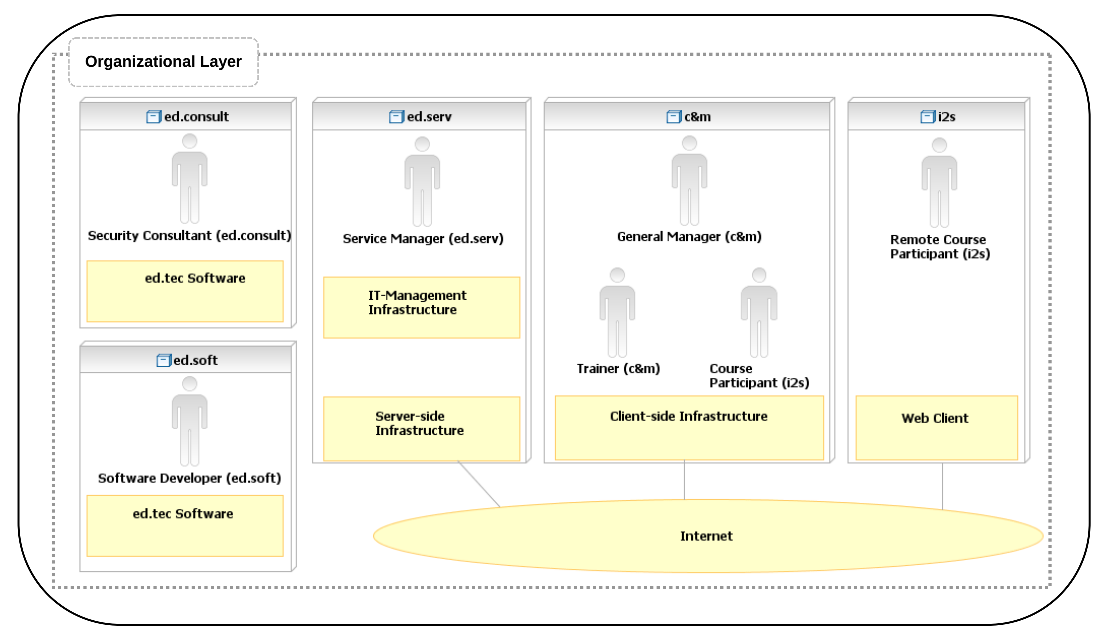
- Entwicklung eines Rollemodells für die rollenbasierte Zugriffskontrolle mit dem Ziel, den Unterschied von Geschäfts- und Systemrollen explizit zu fassen. Durch diese Trennung ergeben sich Möglichkeiten zur Reduktion von Komplexität im produktiven Einsatz, um gesetzlichen Vorgaben entsprechen zu können. Die funktionale Spezifikation des Modells beinhaltet Methoden zur Komplexitätsreduktion.

- Entwicklung eines Vorgehensmodells, welches zur Entwicklung von Geschäfts- und Systemrollen beiträgt. Zusätzlich dazu modelliert das Vorgehensmodell eine explizite Betriebsphase, wodurch der Tatsache Beachtung geschenkt wird, dass ein instanziiertes Rollenmodell im produktiven Einsatz fortwährend Änderungen unterliegt.

- Bewertung aktueller Rollenmanagementwerkzeuge anhand eines in dieser Arbeit entwickelten Kriterienkatalogs. Diese Bewertung findet auf der Basis eines in der Forschungsgruppe C&M entwickelten Evaluierungsszenarios statt.

1.3 Beschreibung des Demonstrators

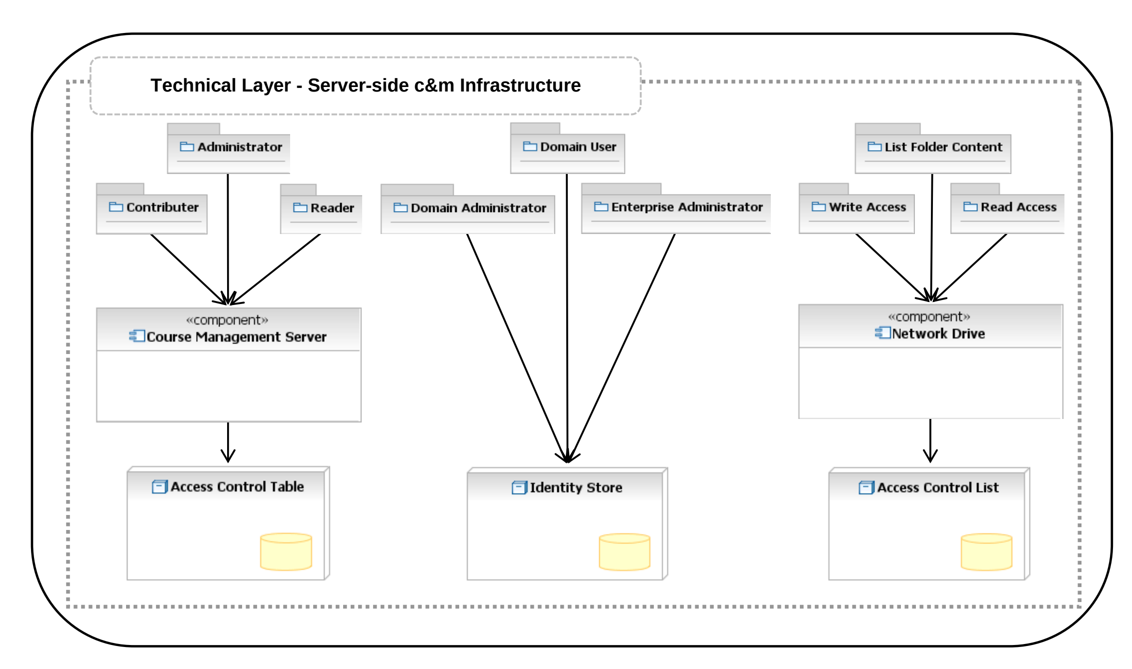
In diesem Teilkapitel wird zunächst ein kurzer Überblick über die Modelle gegeben, die in dieser Arbeit erstellt werden sowie das Szenario für den Tragfähigkeitsnachweis beschrieben. In dem Szenario soll ein vereinfachtes Bild eines Unternehmens skizziert werden, welches im Rahmen einer Fallstudie sowohl den Rollenmodellen, aber auch der Werkzeugevaluation gleichermaßen als Basis dient. Um eine sinnvolle Vergleichbarkeit für beide Rollenwerkzeuge zu ermöglichen, liegt es beiden Betrachtungen zugrunde. Es verfügt über mehrere Geschäfts- und Systemrollen sowie mehrere Systeme und steht somit in direktem Zusammenhang zur Problembeschreibung und der Skizze aus Kapitel 1.2.



Adapted from [C&M-A-BA], page 10

Information 2: Introduction – Business Model in the IST-Scenario

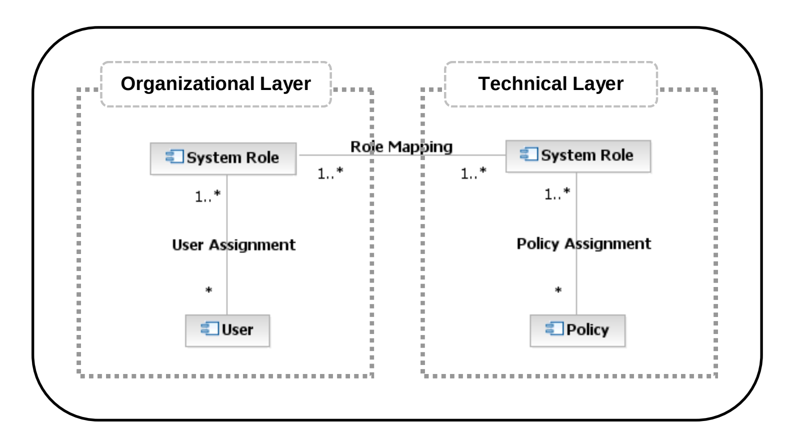
Das Szenario *Internet-Supported Training* (IST) wird in Information 2 dargestellt. Es schildert die Situation eines Anbieters computergestützter Schulungen *cooperation&more* (c&m). Bei c&m können Kursteilnehmer sowohl per Fernzugriff, aber auch vor Ort an Schulungen teilnehmen. Die Netzwerkinfrastruktur von c&m ist dabei aufgeteilt in Arbeitsrechner innerhalb der eigenen Infrastruktur und Serversysteme, die extern verwaltet werden. Das Szenario besteht aus fünf Unternehmen mit unterschiedlichen Geschäftsrollen, die ihrerseits mehrere Mitarbeiter repräsentieren. Die Kompetenz des Unternehmens c&m stellt das Angebot von Schulungen dar und so werden die Anwendungsserver vom IT-Dienstleister ed.serv (*education services*) zur Verfügung gestellt. Die Pflege dieser Systeme wird ebenfalls von ed.serv übernommen. Der Schulungsanbieter c&m besitzt den Kunden *intelligent internet solutions* (i2s), dessen Mitarbeiter entweder über die Arbeitsplätze bei c&m, oder über das Internet per Fernzugriff an Schulungen teilnehmen können. Die Firma ed.consult (*education consulting*) tritt in diesem Szenario als IT-Beratungsfirma auf und integriert die Software-Komponenten der Firma ed.soft (*education software*) in die bestehende Infrastruktur von c&m. Das in diesem Szenario verwendete Schulungsportal von c&m wird als ed.tec (*education technology*) bezeichnet. Diese Partner sowie deren Geschäftsrollen stehen demnach für die Geschäftsebene, die in Information 1 dargestellt wurde und verfügen über unterschiedliche Systemrollen auf technischer Ebene.



Information 3: Introduction – System Model in the IST-Scenario

Information 3 verdeutlicht die technische Ebene aus Information 1 und zeigt das Gesamtbild der c&m-Serverinfrastruktur. Dargestellt sind die Systeme, die von ed.serv verwaltet werden. Die c&m-Infrastruktur verfügt über ein Identitätsmanagementsystem, einen Anwendungsserver für die angebotenen Schulungen sowie einen zentralen Informationsspeicher zum Ablegen von Dokumenten. Jedes dieser Systeme verfügt über eigene Systemrollen, die für unterschiedliche Berechtigungsgrade innerhalb dieser Systeme stehen.

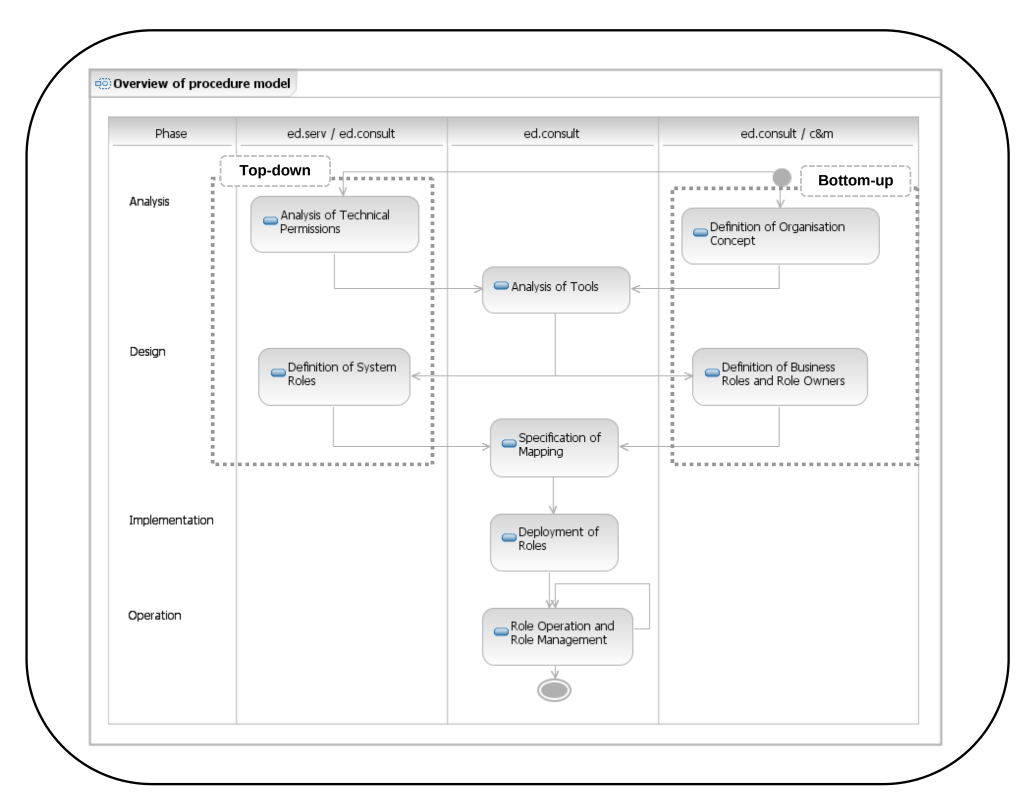
Nachdem ein Überblick über das Szenario gegeben wurde, folgt nun die kurze Vorstellung des Rollenmodells sowie des Vorgehensmodells zum planmäßigen Vorgehen beim Aufbau einer rollenbasierten Unternehmensstruktur. Für die Nachvollziehbarkeit auf Seiten des Lesers wird das Vorgehen anhand des eben eingeführten Szenarios anskizziert und ist in Information 5 dargestellt. Wie in Kapitel 1.2 bereits angedeutet wurde, stellt die Unterscheidung zwischen Geschäfts- und Systemrollen den zentralen Inhalt der Modellspezifikation des Rollenmodells dar. Auf dieser Grundlage werden funktionale Spezifikationen festgelegt, die durch die explizite Trennung der Rollenbegriffe ermöglicht werden. Um einen groben Überblick über das Rollenmodell zu geben, skizziert Information 4 diese explizite Trennung an.



Information 4: Introduction – Role Model

Das Vorgehensmodell orientiert sich am klassischen Software-Entwicklungszyklus und ist unterteilt in die Analysephase, Entwurfsphase, Implementierungsphase und die abschließende Be-

triebsphase. Da ed.consult als Beratungsunternehmen sowohl über technisches Wissen verfügt, als auch die Unternehmensstrukturen von c&m kennt, ist es an dem gesamten Prozess federführend beteiligt. In einem ersten Schritt zur Einführung einer rollenbasierten Unternehmensstruktur ist es nötig, eine Bestandsaufnahme der technischen und organisatorischen Strukturen vorzunehmen. Hier spiegelt sich das eingangs erwähnte hybride Vorgehen wider.



Information 5: Introduction – Procedure Model

In der Aktivität *Analysis of Tools* werden bestehende Werkzeuge aus dem Rollenmanagement analysiert. Diese Arbeit beschäftigt sich in Kapitel 6 mit eben dieser Analyse und greift dabei auf die beiden Werkzeuge Omada Identity Manager und Sun Role Manager zurück. Anschließend werden Geschäfts- und Systemrollen werkzeugunterstützt modelliert. In der abschließenden Phase muss nun ein Zusammenhang zwischen diesen Rollen hergestellt werden. Diese beiden Aktivitäten stehen in Bezug zu den in Kapitel 1.2 erwähnten Begriffen *role discovery* und *role reconciliation*. Abschließend kann die Implementierungsphase in Form einer pilotierten Umsetzung und Tests zur Evaluierung des entwickelten Rollenmodells beginnen. Besondere Beachtung wird im Vorgehensmodell der Phase des Rollenbetriebs (engl. *role operation*) geschenkt, da sich hierdurch der Lebenszyklus eines Rollenmodells im Wirkbetrieb ausdrückt.

1.4 Gliederung der Arbeit

Bisher wurde das Interesse des Lesers auf den Forschungsbereich „Rollenmanagement“ gelenkt und die zentralen Problemstellungen dieser Arbeit formuliert. In diesem Kapitel soll nun eine Übersicht über den Aufbau der vorliegenden Arbeit und eine kurze Beschreibung der einzelnen Kapitel gegeben werden. Im ersten Teil werden die theoretischen Grundlagen für eine nähere Betrachtung des Rollenmanagements gelegt und die beiden Rollenmanagementwerkzeuge Omada Identity Manager und Sun Role Manager vorgestellt. Der zweite Teil befasst sich mit der Entwicklung der Rollenmodelle und nimmt eine Evaluation der Werkzeuge vor. Im letzten Teil dieser Arbeit werden die Modelle im bereits kurz angesprochenen Szenario in einem Rollenmanagementwerkzeug umgesetzt. Abschließend werden die Ergebnisse der Arbeit zentral zusammengefasst und Anknüpfungspunkte für weitere Arbeiten aufgezeigt. Die Zusammenfas-

sung dient somit als Übersicht über die gesamte Arbeit und ihrer entwickelter Konzepte. Auf diese drei Teile wird nun kurz eingegangen.

In Kapitel 2 werden zunächst die Grundlagen für eine detaillierte Betrachtung des Rollenmanagements gelegt. Dazu werden Grundbegriffe erklärt sowie eine Einführung in grundlegende Konzepte, Architekturen und Modelle gegeben. Im Vordergrund steht hierbei eine Vorstellung der im wissenschaftlichen Kontext etablierten Rollenmodelle, die sich unter Anderem mit systemübergreifenden Rollen beschäftigen. In diesem Kapitel wird somit das als gesichert geltende Wissen im Bereich der rollenbasierten Zugriffskontrolle vorgestellt. In Kapitel 3 wird dann ein Überblick über den aktuellen Stand der Technik im Rollenmanagement gegeben und die beiden in dieser Arbeit verwendeten Werkzeuge Omada Identity Manager und Sun Role Manager vorgestellt. Hierbei sind die Ziele, den Marktansatz, die Architekturen sowie den Funktionsumfang beider Lösungen darzustellen. Dieses Kapitel stellt somit aktuelle Ergebnisse aus Wissenschaft und Technik dar, die den zentralen Aspekten dieser Arbeit unmittelbar zugrunde liegen.

In Kapitel 4 wird dann das Rollenmodell BRBAC (engl. *business role-based access control*) aufgestellt. Dazu werden zunächst die zentralen Ziele und Anforderungen erhoben und diese anschließend einzeln umgesetzt. Den Abschluss bildet ein Resümee, welches das Gesamtbild von BRBAC gibt und Anknüpfungspunkte für weitere Forschungsarbeiten nennt. In Kapitel 5 folgt die Entwicklung des Vorgehensmodells für BRBAC. Auch hier werden in analoger Weise zu Kapitel 4 zunächst die Ziele und Anforderungen festgelegt, die dann im Folgenden in den einzelnen Phasen umgesetzt werden. Auf die in den Phasen zu erledigenden Arbeiten wird ebenso eingegangen wie die erzeugten Artefakte jeder Phase, um die Grenze zwischen den Phasen klar zu definieren. Abschließend wird ein Resümee gegeben, welches die zentralen Punkte des Vorgehensmodells zusammenfasst. Kapitel 6 befasst sich mit der Bewertung von Werkzeugen aus dem Rollenmanagement. Diese wurden im ersten Teil der Arbeit bereits wertungsfrei vorgestellt und sollen an dieser Stelle bewertet werden. Dazu wird zunächst ein Kriterienkatalog definiert und die Auswahl der einzelnen Kriterien begründet. Diese stehen unter Anderem in unmittelbarem Zusammenhang zu den beiden in den vorangegangenen Kapiteln entwickelten Modellen.

Im letzten Teil wird zunächst die Tragfähigkeit der Modelle im IST-Szenario belegt und anschließend die gesamte Arbeit zusammengefasst. Zur Demonstration der Tragfähigkeit wird das Szenario in Kapitel 7 zunächst detailliert vorgestellt und eine Auswahl des Rollenmanagementwerkzeugs getroffen. Anschließend wird die Modellinstanziierung von BRBAC gemäß dem entwickelten Vorgehen illustriert und die Trennung der Rollen genauer beleuchtet sowie zum Abschluss die Betriebsphase. Auch für dieses Kapitel werden die wichtigsten Erkenntnisse zusammengefasst ehe im abschließenden Kapitel 8 der Fokus auf die gesamte Arbeit gerichtet wird und die Ergebnisse der gesamten Arbeit zusammengefasst und Anknüpfungspunkte für weitere Arbeiten im Rollenmanagement gegeben werden. Im Anhang befinden sich zusätzliche Informationen, die den Ausführungen der Arbeit zugrunde liegen, den Schwerpunkt aber nicht direkt betreffen.

1.5 Rechtschreibung und Typografie

In dieser Arbeit wird zur Hervorhebung im Text folgendes Schema verwendet:

- *Kursive Schrift* bezeichnet Worte der englischen Sprache, die auch im Fließtext verwendet werden.
- `Schreibmaschinenschrift` wird zur Verwendung von Modellelementen bzw. von Quellcode verwendet.
- Eigennamen sowie Produktnamen werden in dieser Arbeit nicht eigens hervorgehoben, da dies sonst zu einer Verschlechterung des Leseflusses führen würde.

Zusätzlich dazu wird in dieser Arbeit das Wort „Policy“ häufig verwendet. Da dieser Begriff in der Informatik bereits eine klare Bedeutung trägt und die damit einhergehenden Assoziationen

durch die Verwendung der sinngemäßen Übersetzung „Richtlinie“ verloren gingen, wird dieses englische Wort als deutscher Begriff im Fließtext verwendet.

Als Grundlage zur Rechtschreibung in dieser Diplomarbeit wird der aktuelle Duden verwendet [Du06].

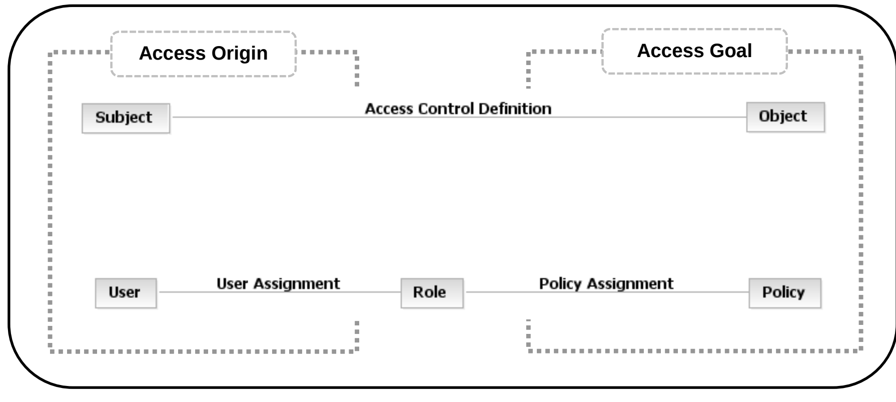
2 GRUNDLAGEN

Diese Arbeit behandelt im Wesentlichen die Modellierung eines Rollen- sowie eines Vorgehensmodells für die rollenbasierte Zugriffskontrolle (engl. *role-based access control*, RBAC). Aus diesem Grund ist es zunächst wichtig, grundlegende Begriffe aus diesem Bereich zum allgemeinen Verständnis einzuführen. Aus diesem Grund wird hier zunächst das grundlegende Konzept der rollenbasierten Zugriffskontrolle vorgestellt, das als alternatives Architekturprinzip zur klassischen Zugriffskontrolle auf der Basis von Identitäten gesehen werden kann (engl. *identity-based access control*, IBAC). Dabei wird RBAC in Bezug gesetzt zur Subjekt/Objekt-Relation, die im Bereich des Identitätsmanagements als abstraktes Modell zur Beschreibung von Zugriffsversuchen verwendet wird [Em08]. Im Anschluss daran wird das RBAC-Standardrahmenwerk eingeführt, welches vier Modelle für die rollenbasierte Zugriffskontrolle mit unterschiedlichem Funktionsumfang definiert. Neben der Entwicklung von Modellen werden in dieser Arbeit sogenannte Rollenmanagementwerkzeuge untersucht und verwendet. Diese verwenden Rollen für die Zugriffskontrolle und beschränken sich dabei nicht auf ein einzelnes technisches Endsystem, sondern weiten die Zugriffskontrolle auf mehrere Systeme aus. Das abschließende Kapitel 2.3 befasst sich daher mit einer Einführung in typische Aufgabenbereiche von Rollenmanagementwerkzeugen.

2.1 Die Subjekt/Objekt-Relation

Die rollenbasierte Zugriffskontrolle stellt ein Modell dar, in dem die Zugriffsberechtigungen, die den Zugriff eines Benutzers auf ein Ziel spezifizieren, explizit vom Benutzerkonto getrennt sind. Um diesen Ansatz zu motivieren, wird hier auf die Subjekt/Objekt-Relation eingegangen und diese in Beziehung gesetzt zum Paradigma der rollenbasierten Zugriffskontrolle.

Die Subjekt/Objekt-Relation kann im Bereich des Identitätsmanagements als Basis für Zugriffskontrollmodelle angesehen werden, wie [Em08] belegt. In Anlehnung daran werden hier die Konzepte „Subjekt“ und „Objekt“ eingeführt. Dabei repräsentiert das Element „Subjekt“ stets ein anfragendes Element beim Zugriff auf ein gewisses Ziel. Das Zugriffsziel wird als „Objekt“ bezeichnet und steht allgemein für jedes vor unbefugtem Zugriff zu schützendes Element. Durch die Verknüpfung zwischen diesen beiden Elementen, die in Form einer Relation ausgedrückt werden kann, wird die Zugriffsberechtigung spezifiziert, die das zugreifende Subjekt auf das zu schützende Objekt besitzt. In der rollenbasierten Zugriffskontrolle wird an diesem Prinzip festgehalten, jedoch wird zwischen Subjekt und Objekt eine Indirektionsstufe eingeführt, die „Rolle“ genannt wird. Durch die Trennung von Subjekt und Objekt wird die Zugriffsberechtigung nun nicht mehr zwischen diesen beiden Elementen definiert, sondern stattdessen zwischen dem neu eingeführten Element „Rolle“ und dem Objekt. Eine zusätzliche Relation wird für die Verknüpfung der Elemente Subjekt und Rolle eingeführt, wodurch die effektive Berechtigung eines Subjekts auf ein Objekt nun insgesamt betrachtet durch zwei Relationen festgelegt wird. Die folgende Abbildung Information 6 stellt den Zusammenhang zwischen der Subjekt/Objekt-Relation und der rollenbasierten Zugriffskontrolle mit dem zusätzlichen Modellelement „Rolle“ her.



Adapted from [Em08]

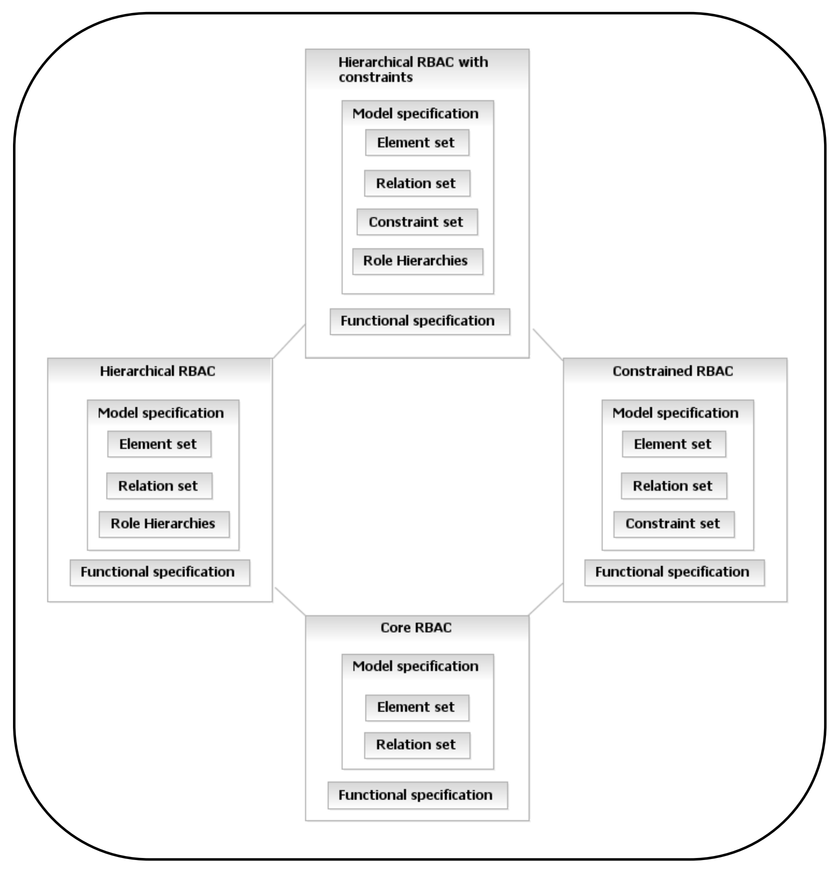
Information 6: Basics – Subject/Object-Relation and RBAC

Ein zentraler Aspekt von RBAC sind neben dem Modellelement „Rolle“ die Relationen Benutzerzuweisung (engl. *user assignment*) und Berechtigungszuweisung (engl. *policy assignment*), durch die eine Rolle einerseits aufgefasst werden kann als semantisches Konstrukt um Berechtigungen, bzw. Policys zu aggregieren und andererseits als Behälter für eine Menge an Subjekten. Durch die explizite Trennung von zugreifenden Subjekten und zugriffsbeschränkten Objekten können Zugriffsrechte (engl: *policies*) nun in konsistenter Form definiert und für eine Menge an Benutzern spezifiziert werden, weil diese mit dem Objekt nun nicht mehr direkt verknüpft sind, sondern in Form der Rolle nur noch indirekt. Ein Benutzer erhält die in einer Rolle definierten Zugriffsrechte, indem er über die Relation *user assignment* in diese Rolle eingeteilt wird. Ein zweiter Vorteil dieses Ansatzes ist, dass Veränderungen an der Zugriffskontrollarchitektur sehr schnell abgebildet werden können, weil diese Änderungen lediglich Änderungen der Rollen, nicht aber jedes einzelnen Benutzers nach sich ziehen. Da Rollen mehrere Benutzer gleichzeitig verkörpern, existieren im Allgemeinen wesentlich weniger Rollen, als Benutzerkonten. Da die Rechte, die in einem System vergeben werden im Allgemeinen diskrete Mengen darstellen und nicht für jeden Benutzer individuell verschieden sind, kann das rollenbasierte Paradigma als wesentliche Verbesserung für Zugriffskontrollarchitekturen angesehen werden. Durch RBAC wird es möglich, Zugriffsrechte in diskreten Mengen und in konsistenter Form zu vergeben, was die Effizienz in der Rollenverwaltung potentiell erhöht.

Auf dieser Basis wird im folgenden Kapitel mit NIST-RBAC ein Standardrahmenwerk für rollenbasierte Zugriffskontrollarchitekturen eingeführt, welches das RBAC-Paradigma in Modellen spezifiziert, die aufeinander aufbauen und sich im Funktionsumfang voneinander unterscheiden. Dieser Standard bildet die Grundlage für das ERBAC-Rollenmodell, welches in Kapitel 3.1.2 eingeführt wird. Diese beiden Modelle bilden gleichermaßen das Fundament für die Entwicklung des Rollenmodells in Kapitel 4.

2.2 Das NIST-RBAC-Standardrahmenwerk

In diesem Teilkapitel wird der NIST-RBAC-Standard vorgestellt und die in diesem Rahmenwerk definierten Begriffe und Konzepte eingeführt. NIST-RBAC führt verschiedene konzeptionelle Komponentenmodelle ein, die für die Betrachtung in dieser Arbeit elementar sind. Zunächst wird ein Überblick über das gesamte Rahmenwerk gegeben und anschließend auf die darin definierten Komponenten separat eingegangen. Jede Komponente des RBAC-Standards ist in die zwei Spezifikationsklassen „Referenzmodellspezifikation“ und „funktionale Spezifikation“ aufgeteilt. Im Referenzmodell werden die Modellelemente und Relationen spezifiziert (engl. *model specification*), während die funktionale Spezifikation (engl. *functional specification*) administrative und systematische Funktionen im Bezug auf die Modellelemente betrachtet [FK+07, FS+01]. Information 7 gibt einen Überblick über die vier Komponentenmodelle, die im RBAC-Standard enthalten sind und stellt für jedes Modell die beiden Spezifikationsklassen dar.



Information 7: Basics – NIST-RBAC Reference Models

Wie aus der Abbildung hervorgeht, besteht der NIST-RBAC-Standard aus den vier Modellen *core RBAC*, *hierarchical RBAC*, *constrained RBAC* und *hierarchical RBAC with constraints*, die aufeinander aufbauen und vier unterschiedlich komplexe Modelle darstellen. In den folgenden Teilkapiteln wird jedes dieser vier Modelle vorgestellt. Dabei wird zunächst das jeweilige Modell selbst spezifiziert und anschließend auf dessen Funktionen eingegangen. An dieser Stelle sei erwähnt, dass der NIST-RBAC-Standard seinerseits stark von RBAC96 geprägt ist und wesentliche Konzepte daraus verwendet. In dieser Arbeit werden das Zusammenspiel, Gemeinsamkeiten oder Gegensätze dieser beider Rahmenwerke allerdings nicht explizit diskutiert, da sie selbst lediglich die Basis für diese Arbeit darstellen. Eine derartige Diskussion würde am Kern dieser Arbeit nichts ändern und deshalb wird hier lediglich verwiesen auf [SB08].

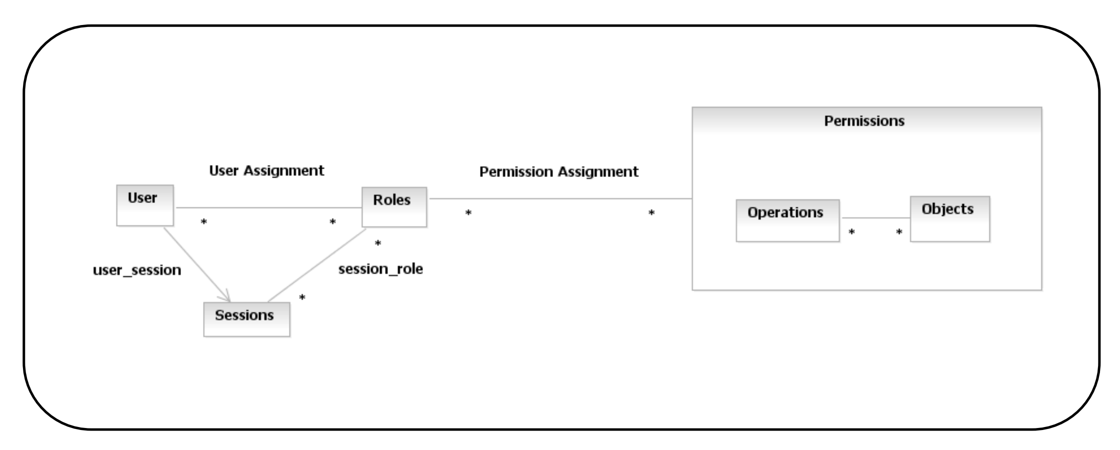
2.2.1 RBAC Kernmodell

Wie aus Information 7 hervorgeht, stellt das *core RBAC*-Modell das Basismodell im NIST-RBAC-Standard dar. Es spezifiziert einerseits die grundlegenden Modellelemente für RBAC und andererseits die dafür benötigten Funktionen. Im Folgenden wird nun zunächst auf die Modellelemente und anschließend auf die funktionale Spezifikation eingegangen.

Modellspezifikation

Das *core RBAC*-Modell besteht aus fünf Elementen zur Modellierung von RBAC. Dies sind Benutzer (engl. *users*), Rollen (engl. *roles*), Objekte (engl. *obejcts*), Operationen (engl. *operations*) und Berechtigungen (engl. *permissions*). Durch das Modellelement *roles* werden Benutzer und Objekte in eine m-zu-n-Beziehung zueinander gestellt, wie in Kapitel 2.1 über die Subjekt/Objekt-Relation für RBAC bereits angedeutet wurde. Dabei stellen m und n die Kardinalitäten dieser beider Elemente dar. Mit dem Modellelement Sitzung (engl. *session*) wird der aktuelle Kontext festgelegt, in dem gearbeitet wird. Mit der Hilfe von Sitzungen werden Rollen

kontextsensitiv und eine Teilmenge an Rollen selektiert, die nur im vorliegenden Kontext aktiviert werden können. Modelliert wird dieser Sachverhalt durch die Relation *session_role*. Ein Benutzer bezeichnet in diesem Zusammenhang immer einen menschlichen Benutzer, obwohl ein Benutzerkonto im Allgemeinen ja auch für computersystemeigene Benutzerkonten stehen kann. Der NIST-RBAC-Standard definiert einen Benutzer jedoch stets als menschlichen Benutzer. Unter einer Rolle versteht das Standardrahmenwerk nunmehr eine Jobfunktion innerhalb einer Organisation oder Einrichtung. Semantisch gesehen steht eine Rolle somit für eine Arbeitsaufgabe, die einem Benutzer über die Relation *user assignment* (UA) zuteil wird, wenn er in die entsprechende Rolle eingeteilt wird. Unter einer *permission* wird die Berechtigung verstanden, die zu dieser Arbeitsaufgabe gehört und die das Subjekt in Form seiner Rolle beim Zugriff auf ein Objekt besitzt. Dabei besteht eine *permission* selbst aus einer Menge an Operation, die systemspezifisch sind und den Zugriff auf ein Objekt festlegen. Im Allgemeinen stellt die Menge an Operationen demnach ausführbare Programmeinheiten dar, welche im Kontext des Benutzers gewisse Funktionen ausführen. Diese Operationen hängen dabei von dem System ab, in dem sie definiert sind und können daher sehr unterschiedlich sein.



Adapted from [FS+01], Figure 1

Information 8: Basics – NIST-RBAC *core RBAC*

Information 8 stellt die Modellspezifikation von *core RBAC* grafisch dar. Die Elemente und Relationen sind hierbei folgendermaßen definiert:

- Modellelementmengen: *users*, *roles*, *operations*, *objects*, *sessions*, *permissions*
- *permissions* = $2^{(\text{operations} \times \text{objects})}$, Menge an Berechtigungen.
- UA $\subseteq$ *users* × *roles*, m-zu-n-Beziehung zwischen Benutzern und Rollen.
- PA $\subseteq$ *permissions* × *roles*, m-zu-n-Beziehung zwischen Rollen und Berechtigungen.
- Op(p: *permissions*) $\rightarrow$ {op $\subseteq$ *operations*}, Abbildung von Berechtigungen auf Operationen; dient zur Rückgabe der Operationen, für die die Berechtigung definiert ist.
- Ob(p: *permissions*) $\rightarrow$ {ob $\subseteq$ *objects*}, Abbildung von Berechtigungen auf Objekte; dient zur Rückgabe der Objekte, die mit der *permission* belegt sind.
- *user_sessions* (u: *user*) $\rightarrow 2^{\text{session}}$, Abbildung eines Benutzers auf dessen Sitzungen.
- *assigned_users* (r: *roles*) $\rightarrow 2^{\text{users}}$, Abbildung einer Rolle auf die darin definierten Benutzer.
  assigned_users (r) = {u $\in$ *users* | (u,r) $\in$ UA};
- *assigned_permissions* (r: *roles*) $\rightarrow 2^{\text{permissions}}$, Abbildung einer Rolle auf eine Teilmenge der Berechtigungen.
  assigned_permissions (r) = {p $\in$ *permissions* | (p,r) $\in$ PA};

- *session_roles* (s: *session*) $\rightarrow 2^{roles}$, Abbildung einer Sitzung auf die darin definierten Rollen.
  session_roles (s_i) $\subseteq$ {r $\in$ *roles* | (*session_users* (s_i), r) $\in$ UA};
- *avail_session_perms* (s: *sessions*) $\rightarrow 2^{permissions}$, Abbildung eines Benutzers auf Berechtigungen für eine spezielle Sitzung.
  avail_session_perms (s) = { $\bigcup_{r \in session_roles(s)} assigned_permissions(r)$ };

Funktionale Spezifikation

Nachdem das Referenzmodell *core RBAC* beschrieben wurde, wird im Folgenden auf dessen funktionale Spezifikation eingegangen. Diese Spezifikation befasst sich mit Funktionen im RBAC-Kontext, die in administrative Funktionen, systemunterstützende Funktionen und Funktionen zur Überarbeitung der Rollen unterteilt werden.

- **Administrative Funktionen.** Diese befassen sich mit dem Erzeugen und Verwalten der Modellelemente und Relationen aus der Modellspezifikation von *core RBAC*. Die grundlegenden Mengen zur Administration in RBAC sind *users*, *roles*, *operations* und *objects*. Da die Mengen *operations* und *objects* technische Endsysteme darstellen, liegt deren Administration außerhalb von RBAC. Für die verbleibende Menge existieren die administrativen Operationen `addUser` und `deleteUser` im Falle von Benutzern und `addRole` und `deleteRole` für Rollen. Zur Verwaltung der beiden wichtigsten Relationen *user assignment* (UA) und *permission assignment* (PA) existieren Folgende administrative Funktionen: Für UA werden `assignUser` und `deassignUser` definiert und für PA existieren `grantPermission` und `revokePermission`.

- **Systemunterstützende Funktionen.** Diese Spezifikation richtet sich an das Verwalten von Sitzungskontexten sowie der Entscheidungsfindung zur Beantwortung von Zugriffskontrollanfragen. Um Aussagen über den Zugriff auf ein Objekt machen zu können, wird eine im aktuellen Sitzungskontext aktive Rolle benötigt, die ihrerseits über Benutzer verfügt. Diejenige Funktion, die einen Sitzungskontext eröffnet, stellt einem Benutzer gleichzeitig die Teilmenge an Rollen zur Verfügung, die in diesem Kontext gültig sind. Diese Menge kann dann während der Sitzung durch den Benutzer selbst geändert werden, indem Rollen hinzugenommen oder entfernt werden. Folgende Funktionen stehen an dieser Stelle zur Verfügung:
  - `createSession`: Erzeugt eine Benutzersitzung und berechnet die Menge an zur Verfügung stehenden Rollen.
  - `addActiveRole`: Fügt eine Rolle als aktive Rolle für diesen Kontext hinzu.
  - `dropActiveRole`: Entfernt eine Rolle aus dem für diese Sitzung aktiven Rollenvorrat.
  - `checkAccess`: Liefert einen Wahrheitswert zurück, der darüber Aussagen macht, ob das zugreifende Subjekt im aktiven Kontext die gewünschte Operation auf einem Objekt ausführen darf, oder nicht.

- **Funktionen zur Überarbeitung.** Diese Funktionen richten sich nicht an einen Benutzer, der seine im aktiven Kontext vorhandenen Rollen beeinflussen will, sondern an einen Administrator, dem es möglich sein sollte, alle aktuellen Zuweisungen zu überprüfen. Diese Funktionen richten sich insbesondere an die Relationen UA und PA. Da dies nicht von allen RBAC-Implementierungen unterstützt wird, unterteilt der RBAC-Standard diese Funktionen in zwingend benötigte und optionale Funktionen. Zwingend (Z) bzw. optional (O) vorhanden sind folgende Funktionen:
  - `assignedUser`: Liefert die Menge an Benutzern zurück, die in eine bestimmte Rolle eingeteilt sind (Z).

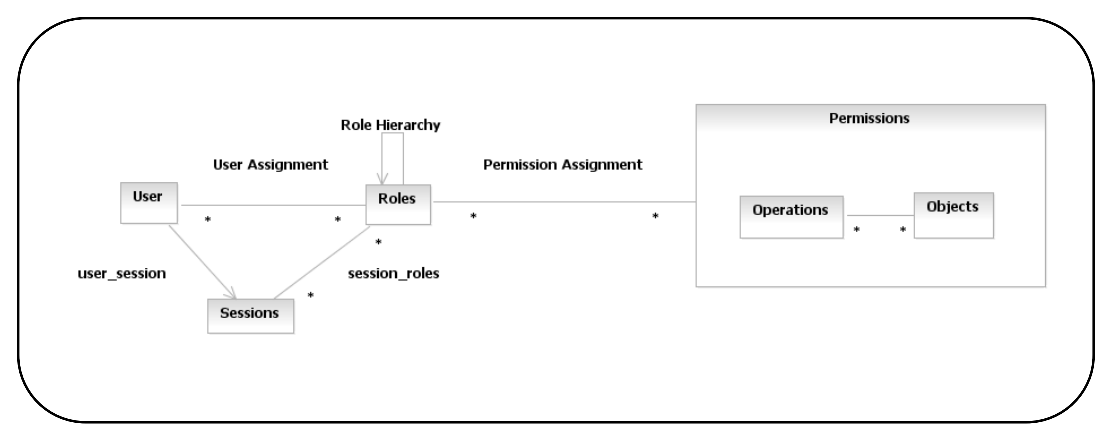
- `assignedRoles`: Liefert die Menge an Rollen zurück, über die ein bestimmter Benutzer verfügt (Z).
- `rolePermissions`: Liefert die Menge an Berechtigungen für eine bestimmte Rolle zurück (O).
- `userPermissions`: Liefert die Menge an Berechtigungen zurück, über die ein Benutzer durch seine Rollenmitgliedschaften direkt oder transitiv verfügt (O).
- `sessionRoles`: Liefert die Teilmenge der in einem Sitzungskontext aktiven Rollen eines Benutzers zurück (O).
- `sessionPermissions`: Liefert die Teilmenge der in einem Sitzungskontext aktiven Berechtigungen zurück (O).
- `roleOperationsOnObject`: Liefert die Menge an Operationen zurück, über die eine bestimmte Rolle beim Zugriff auf ein bestimmtes Objekt verfügt (O).
- `userOperationsOnObject`: Liefert die Menge an Operationen zurück, über die ein bestimmter Benutzer beim Zugriff auf ein bestimmtest Objekt verfügt (O).

2.2.2 RBAC-Modell mit Hierarchien

Das Modell *hierarchical RBAC* erweitert *core RBAC* um Hierarchien auf der Basis von Rollen. Alle in *core RBAC* definierten Spezifikation sind gleichermaßen gültig, müssen jedoch durch die Einführung von Hierarchien gegebenenfalls abgeändert werden. Im Folgenden wird nun wieder zunächst auf die Modellspezifikation und im Anschluss daran auf die funktionale Spezifikation eingegangen.

Modellspezifikation

Zur Modellierung von Hierarchien wird eine zusätzliche Relation *role hierarchy* eingeführt, die auf Rollen definiert ist. Hierarchien sind im Zusammenhang von RBAC ein wichtiger Aspekt, so dass weiterführende Arbeiten, wie etwa Implementierungen von Rollenmodellen in kommerziellen Produkten meist auf diesem Modell, oder auf *hierarchical RBAC with constraints* aufbauen, welches im nächsten Teilkapitel besprochen wird. Hierarchien beschreiben eine Vererbungsrelation, die auf Rollen beschränkt ist. Dabei erbt eine Rolle r_1 von einer Rolle r_2 alle in ihr aufgeführten Berechtigungen (engl. *permissions*). Im RBAC-Standard werden für Hierarchien sowohl partielle als auch strenge Ordnungen definiert. Das führt dazu, dass es sowohl möglich ist, dass eine Rolle über genau einen Vorgänger und beliebig viele Nachfolgerrollen verfügen kann. Ebenso ist es möglich, genau einen Nachfolger und beliebig viele Vorgänger zu modellieren. In der Hierarchie kann demnach eine Rolle beliebig viele Vorgänger und Nachfolger besitzen. Veranschaulichen kann man diese Rollenhierarchien durch Bäume zur Modellierung genau eines Vorgängers mit mehreren Nachfolgern, invertierte Bäume zur Modellierung genau eines Nachfolgers mit beliebigen Vorgängern oder als Gitter zur Modellierung einer Kombination aus beidem. Diese differenzierte Betrachtung unterschiedlicher Hierarchieformen wird nicht näher beleuchtet.



Adapted from [FS+01], Figure 2

Information 9: Basics – NIST-RBAC *hierarchical RBAC*

Diese Abbildung zeigt das Rollenmodell mit Hierarchien, die auf der Basis von Rollen modelliert werden. Zusätzlich zu den Spezifikationen für das *core RBAC*-Modell definiert hierarchisches RBAC folgende Elemente und Relationen:

- `Authorized_users (r: roles)` $\rightarrow 2^{users}$, Abbildung einer Rolle r auf eine Menge von Benutzern, die entweder direkt in die Rolle eingeteilt sind, oder durch die Rollenhierarchie enthalten sind.
  `Authorized_users(r)` $= \{u \in \text{users} \mid r' \succeq r, (u, r') \in UA\}$;
- `Authorized_permissions (r: roles)` $\rightarrow 2^{permissions}$, Abbildung einer Rolle r auf eine Menge von Berechtigungen, über die die Rolle entweder direkt, oder durch die Vererbung entlang der Hierarchie verfügt.
  `Authorized_permissions (r)` $= \{p \in \text{permissions} \mid r' \succeq r, (p, r') \in PA\}$;
- *Role hierarchy* (RH): $RH \subseteq \text{roles} \times \text{roles}$.
  Dabei beschreibt RH eine partielle Ordnung auf Rollen. Dabei gilt:
  $r_1 \preceq r_2 \Rightarrow \textit{authorized_permissions}(r_2) \subseteq \textit{authorized_permissions}(r_1) \wedge \textit{authorized_users}(r_1) \subseteq \textit{authorized_users}(r_2)$;
  Im Fall von beschränkten Hierarchien, wo pro Rolle maximal ein Nachfolger möglich ist, gilt zusätzlich folgende Einschränkung:
  $\forall r, r_1, r_2 \in \text{roles}, r \succeq r_1 \wedge r \succeq r_2 \Rightarrow r_1 = r_2$;

Funktionale Spezifikation

Nachdem das hierarchische RBAC-Modell spezifiziert ist, folgt nun eine Auflistung der vorhandenen funktionalen Spezifikation, aufgeteilt in die drei Bereiche administrativer Funktionen, Funktionen zur Systemunterstützung und Überarbeitungsfunktionen:

- **Hierarchische, administrative Funktionen.** Zusätzlich zu den administrativen Funktionen, die in *core RBAC* festgelegt wurden, muss die Möglichkeit gegeben werden, auf die Hierarchiebildung einzugehen. Im administrativen Kontext sind dies Funktionen, die Änderungen an der Hierarchie vornehmen. An dieser Stelle muss erwähnt werden, dass die aus *core RBAC* bekannten Funktionen zur Administration `assignUser`, `deassignUser`, `assignPermission` und `deassignPermission` abgeändert werden müssen, damit sie auch auf Hierarchien operieren. Änderungen an der Benutzer/Rolle-Relation und Rolle/Berechtigung-Relation wirken sich nun nicht mehr nur auf die angegebene Rolle selbst aus, sondern zusätzlich dazu auf die vererbten Rollen. Wird etwa ein Benutzer in eine Rolle r_1 eingetragen, erhält er zusätzlich zu r_1 alle Rollen, die r_1 selbst erbt. Die administrativen Funktionen, die auf Hierarchien operieren, sind:

- `addInheritance`: Erzeugt eine direkte Vererbungsbeziehung zwischen zwei existierenden Rollen.
- `deleteInheritance`: Hebt eine Vererbungsbeziehung zwischen zwei Rollen auf.
- `addAscendent`: Erzeugt eine neue Rolle und fügt diese in die Hierarchiestruktur als unmittelbarer Vorgänger zu einer existierenden Rolle ein.
- `addDescendent`: Erzeugt eine neue Rolle und fügt diese in die Hierarchiestruktur als unmittelbarer Nachfolger zu einer existierenden Rolle ein.

An dieser Stelle sei ergänzend erwähnt, dass die spezifizierte Hierarchiestruktur bei der Implementierung dieser Funktionen beachtet werden muss. Sollte eine Hierarchie modelliert worden sein, die nur einen unmittelbaren Vorgänger erlaubt, wie es etwa durch einem Baum dargestellt werden kann, kann eine Rolle über die Funktion `addAscendent` nur dann eine Vorgängerrolle erhalten, wenn sie bisher über keine verfügt, oder aber der bisherige Vorgänger überschrieben wird.

- **Systemunterstützende Funktionen.** Die Funktionen zur Unterstützung der Systeme sind identisch mit den Funktionen aus *core RBAC*, jedoch wirkt sich die Hierarchie auf die Funktionsweise von `createSession` und `addActiveRole` aus. Durch `createSession` wird die Menge an aktiven Rollen in einem vorgegebenen Kontext berechnet, während durch `addActiveRole` der Benutzer einer Rolle für den aktuellen Kontext aktiviert werden kann. Es stellt sich die Frage, ob hierdurch nur die explizit angegebenen Rollen aktiviert werden sollen, oder auch diejenigen Rollen, die sich aus der Vererbungsbeziehung ergeben. Der NIST-RBAC-Standard sieht hierfür vor, dass bei der initialen Bestimmung der für einen speziellen Kontext gültigen Rollen durch `createSession` sowohl die direkten, als auch die vererben Rollen aktiviert werden. Bei der zusätzlichen Aktivierung weiterer Rollen durch `addActiveRole` hingegen werden nur die Rollen, die der Benutzer selbst auswählt, beachtet, ohne dabei die Vererbungen mit einzubeziehen.

- **Funktionen zur Überarbeitung.** Die Überarbeitungsfunktionen aus *core RBAC* sind auch hier weiterhin gültig. Durch die Definition von Hierarchien enthält die Menge der Benutzer in einer Rolle nun neben den direkt mit der Rolle verknüpften Benutzern zusätzlich noch diejenigen Benutzer, die durch die Hierarchie vererbt werden. Ebenso besteht die Rollenmitgliedschaft im Benutzerobjekt nun nicht mehr nur aus den explizit vergebenen Rollen, sondern darüber hinaus auch aus vererbten Rollen. Um dies sicherzustellen, werden folgende Funktionen definiert:
  - `authorizedUser`: Liefert die Menge der direkt und indirekt in einer Rolle vorhandenen Benutzer zurück.
  - `authorizedRoles`: Liefert die Menge an Rollen zurück, die direkt oder indirekt in einem Benutzerobjekt vorhanden sind.

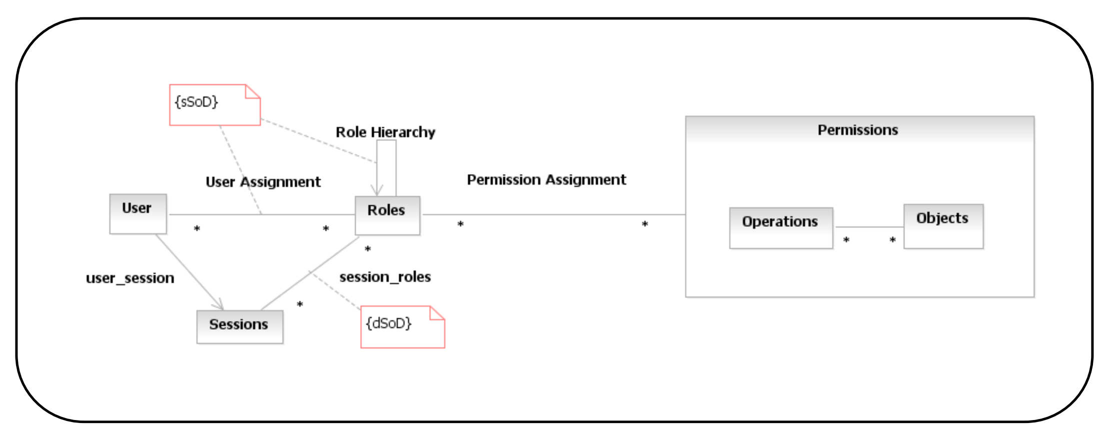
Neben den Änderungen, die die Relation Benutzer/Rolle betreffen, müssen auch diejenigen Funktionen angepasst werden, die auf den bereits erteilten Berechtigungen für Rollen operieren. Durch die Rollenhierarchie unterliegen auch die Berechtigungen dem Vererbungsmechanismus. Wie die Überarbeitungsfunktionen in *core RBAC* bereits erwähnen, gelten folgende vier Funktionen als optional, weil sie nicht in allen RBAC-Umsetzungen vorhanden sind:
  - `rolePermissions`: Liefert die Menge an Berechtigungen zurück, die direkt oder indirekt an eine Role vergeben wurden.
  - `userPermissions`: Liefert die Menge an Berechtigungen eines gegebenen Benutzers zurück, die er durch direkte und indirekte Rollenmitgliedschaften erhält.

- `roleOperationsOnObject`: Liefert die Menge an Operationen zurück, die eine Rolle durch explizite oder implizite Berechtigungen an einem Objekt ausführen darf.
- `userOperationsOnObject`: Liefert die Menge an Operationen zurück, die ein Benutzer an einem gegebenen Objekt ausführen darf, unter Beachtung dessen direkten und indirekten Rollenmitgliedschaften.

2.2.3 RBAC-Modell mit Beschränkungen

Das *constrained RBAC*-Modell erweitert den NIST-RBAC-Standard um die Möglichkeit, Beschränkungen (engl. *constraints*) für Relationen zu definieren. Wie aus Information 7 hervorgeht, stellt *constrained RBAC* keine Erweiterung des in Kapitel 2.2.2 eingeführten hierarchischen RBAC-Modells dar, sondern basiert in gleicher Weise auf *core RBAC*. Im Gegensatz zu Hierarchien führt dieses Modell Beschränkungen als Erweiterung ein, um zu ermöglichen, dass die Modellelemente nur unter gewissen Umständen zueinander in Relation gestellt werden können. Dies wird etwa dazu verwendet, *separation of duty* (SoD) zu modellieren. Dieses Prinzip verkörpert den wechselseitigen Ausschluss für Berechtigungen, etwa um es Benutzern unmöglich zu machen, das in ihrer Jobfunktion definierte Maß an Rechten zu überschreiten. Da Berechtigungen in Rollen gekapselt sind, operieren *separation of duty*-Beschränkungen nicht auf einzelnen Berechtigungen, sondern auf Rollen. Dieser Ansatz führt somit dazu, dass eine Überschreitung von Berechtigungen innerhalb einer Organisation durch die Zugriffskontrollarchitektur vermieden wird und lediglich außerhalb dessen Einflussbereichs auftreten kann. Ein Beispiel hierfür sind Verstöße, die zwischen Mitarbeitern direkt kommuniziert werden. Die Policy-Konformität stellt einen wichtigen Aspekt in Zugriffskontrollarchitekturen dar, wie unter anderem [SB08] belegt, was die Bedeutung von SoD-*constraints* beim Einsatz einer rollenbasierten Zugriffskontrollarchitektur hervorhebt. Im *constrained RBAC*-Modell werden zwei unterschiedlichen Ausprägungen von SoD modelliert, auf die im Folgenden detailliert eingegangen wird. Dabei handelt es sich um statisches SoD zur Definition des wechselseitigen Ausschlusses von Rollen und dynamisches SoD, das diesen Ausschluss im gleichen Kontext betrachtet.

Der NIST-RBAC-Standard definiert zur Spezifikation von *constraints* zwei eigene Rollenmodelle. Das *constrained RBAC*-Modell schließt die Verwendung von Hierarchien explizit aus, während *hierarchical RBAC with constraints* beide Erweiterungen in einem Rollenmodell spezifiziert. Der Zusammenhang zwischen diesen vier Rollenmodellen aus NIST-RBAC ist in Information 7 aufgezeigt. Da RBAC-Modelle, die keine Hierarchien aufweisen, sowohl in der Praxis, als auch bei den weiteren Betrachtungen in dieser Arbeit nicht von Bedeutung sind, wird das Modell *constrained RBAC* nicht explizit betrachtet. Stattdessen soll der Fokus auf dem mächtigsten dieser vier Modelle liegen, dem Modell *hierarchical RBAC with constraints*. Im Folgenden werden nun zunächst die beiden Typen von SoD eingeführt, wie sie in diesem Modell definiert sind und anschließend auf die funktionale Spezifikation dieses Modells eingegangen.



Adapted from [FS+01], Chapter 3.3

Information 10: Basics – NIST-RBAC *constrained RBAC with hierarchies*

Modellspezifikation von statischem *separation of duty*

Konflikte im Zusammenhang mit Rechten, über die ein Benutzer verfügt, werden in einer rollenbasierten Zugriffskontrollarchitektur dadurch verursacht, dass ein Benutzer in Rollen eingeteilt ist, die ihrerseits Rechte subsumieren, die sich gegenseitig ausschließen. Eine Möglichkeit, diesen Sachverhalt zu vermeiden, ist die Definition von statischen SoD-*constraints*. Der RBAC-Standard sieht dazu die Definition von Beschränkungen für die Relation Benutzer/Rolle vor. Das bedeutet, dass bei der Einteilung eines Benutzers in eine Rolle r_1 überprüft werden muss, ob eine Beschränkung für r_1 definiert wurde. Sollte der Benutzer bereits über eine Rolle r_2 verfügen, die sich mit r_1 wechselseitig ausschließt, kann die Einteilung nicht durchgeführt werden. Im Allgemeinen stellt ein SoD-*constraint* somit eine Menge an Rollen dar, von denen ein Benutzer maximal eine besitzen darf. Der NIST-RBAC-Standard nennt zwei Arten, wie statisches SoD technisch umgesetzt werden kann: Entweder verfügt jede Rolle über eine eigene Liste, oder es existiert eine global vorgehaltene Liste mit den Rollen, die sich wechselseitig ausschließen. Wenn sich die Mitglieder einer Rolle ändern, muss nun jedes Mal geprüft werden, ob ein statisches SoD-*constraint* vorliegt, um die Policy-Konformität sicherzustellen. Im hier beschriebenen NIST-RBAC-Rollenmodell wird statisches SoD als globales Element zusammen mit Kardinalitäten definiert. Durch die Angabe einer Kardinalität n für ein SoD-*constraint* wird definiert, über wie viele Rollen aus dieser Menge ein Benutzer verfügen kann, ehe eine Policy-Verletzung vorliegt. Das Rollenmodell *hierarchical RBAC with constraints* definiert statisches SoD (sSoD) folgendermaßen:

- $sSoD \subseteq (2^{roles} \times n), n \in \mathbb{N}, n \geq 2;$
  sSoD ist demnach eine Menge von Tupeln (rs,n) mit rs $\subseteq$ roles, so dass kein Benutzer über mehr als n Rollen aus rs verfügt.
  $sSoD(n) = \{(rs,n) \mid \forall (rs,n), \forall\ t \subseteq rs: |t| \geq n \Rightarrow \bigcap_{r \in t} assigned_users(r) = \emptyset\};$

Modellspezifikation von dynamischem *separation of duty*

Durch Relationen mit statischen SoD-*constraints* verringert sich die Anzahl potentiell zur Verfügung stehender Rollen, in die ein Benutzer eingeteilt werden kann. Auch dynamisches SoD verfolgt das Ziel, die Anzahl der zur Verfügung stehenden Rollen zu verringern, jedoch wird durch dynamische SoD-*constraints* zusätzlich zur rein statischen Betrachtung der Kontext als zusätzlich beschränkender Parameter hinzugenommen. Statisches SoD beschränkt die Dimension der zur Verfügung stehenden Berechtigungen durch eine Beschränkung aller potentiell zur Verfügung stehenden Rollen, wohingegen dynamisches SoD diese Beschränkung auf der Rollenteilmenge im aktuellen Kontext vornimmt. Die Art und Weise, wie der RBAC-Standard dSoD modelliert, ermöglicht, dass gewisse Rechte genau für den Zeitraum bei einem Benutzer vorhanden sind, in der er im selben Kontext auftritt, nicht jedoch darüber hinaus. Dieses Prinzip

ist bekannt als *timely revocation of trust*. Für eine vertiefende Betrachtung verschiedener SoD-Typen sei an dieser Stelle verwiesen auf [FK+07, Kapitel 5.1]. An dieser Stelle ist es wichtig, festzustellen, dass durch sSoD erreicht wird, dass ein Benutzer über keine Rechte verfügt, die sich aufgrund interner Policys oder rechtlicher Vorgaben ausschließen. Dynamisches SoD hingegen ermöglicht die Definition von Rollen, die sich nicht prinzipiell wechselseitig ausschließen, sondern lediglich, wenn sie im gleichen Kontext zur Ausführung gebracht werden.

- $dSoD \subseteq (2^{roles} \times n), n \in \mathbb{N}, n \geq 2;$
  dSoD ist eine Menge von Tupeln (rs,n) mit $rs \subseteq roles$, so dass kein Benutzer mehr als n Rollen aus rs in jedem Element rs aktivieren kann.
  $dSoD(n) = \{(rs,n) \mid \forall rs_1, rs_2 \in 2^{roles}, \forall s \in session, n \in N: n \geq 2 \wedge |rs_1| \geq n \wedge rs_2 \subseteq rs_1, rs_2 \subseteq session_roles(s) \Rightarrow |rs_1| < n\}$

Verfügt das Rollenmodell neben *constraints* auch noch über Hierarchien, müssen die eben erwähnten Definitionen für sSoD und dSoD angepasst werden. Für sSoD wird demnach die Menge an Rollen substituiert durch diejenigen Rollen, die einerseits direkt vorhanden sind wie bisher sowie diejenigen Rollen, die durch die Vererbung entlang der Hierarchie hinzukommen. Die wird durch die Funktion `authorized_users` zum Ausdruck gebracht.

- $sSoD(n) = \{(rs,n) \mid \forall (rs,n), \forall t \subseteq rs: |t| \geq n \Rightarrow \bigcap_{r \in t} authorized_users(r) = \emptyset\}$

Da nun die Modellspezifikation für RBAC mit SoD-*constraints* sowie Hierarchien definiert wurde, folgt nun die funktionale Spezifikation für sSoD und dSoD.

Funktionale Spezifikation für die statische SoD-Relation

- **Administrative Funktionen mit *constraints*.** Im Falle von *constrained RBAC* ohne Hierarchien beinhalten die administrativen Funktionen alle in *core RBAC* definierten Funktionen. Da jedoch die Definition von sSoD auf Benutzermitgliedschaften operiert, deren Rollen sich wechselseitig ausschließen, muss die Funktion `assignUser` insofern erweitert werden, als dass sie überprüft und sicherstellt, dass eine Benutzerzuweisung keine Policy-Verletzung verursacht. In diesem Referenzmodell wird ein SoD-*constraint* als 3-Tupel aufgefasst, das aus einer Referenz auf einen Geschäftsprozess besteht, für den die Beschränkung gelten soll, einer Menge an Rollen, die dabei zueinander in Konflikt stehen und der Kardinalität, die darüber Aussagen macht, über wie viele Rollen aus der Menge ein Benutzer gleichzeitig verfügen darf, ehe eine Policy-Verletzung auftritt. Aus diesem Grund sind administrative Funktionen für die Erzeugung bzw. Löschung von SoD-*constraints* solche Operationen, die eine derartige Instanz anlegen bzw. löschen, oder Änderungen an den drei beschriebenen Parametern vornehmen. Folgende Operationen werden definiert:
  - `createSSoDSet`: Erzeugen einer benannten Instanz eines statischen SoD-*constraint*.
  - `deleteSSoDSet`: Löschen einer existierenden statischen SoD-Relation.
  - `addSSoDRoleMember`: Hinzufügen einer Rolle zu einer bestehenden sSoD-Instanz.
  - `deleteSSoDRoleMember`: Entfernen einer Rolle aus einer statischen SoD-*constraint*-Instanz.
  - `setSSoDCardinality`: Ändern der Kardinalität dieser Instanz.

- **Systemunterstützende Funktionen.** Die Funktionen zur Unterstützung von Systemen sind identisch mit denen aus *core RBAC*.

- **Funktionen zur Überarbeitung.** Für die Implementierung des statischen SoD-Modells sind alle Funktionen zur Überarbeitung aus *core RBAC* nötig. In Anlehnung an die zu-

sätzlichen administrativen Funktionen sind an dieser Stelle ebenfalls weitere Funktionen nötig, die die Ergebnisse der administrativen Funktionen für die Überarbeitung des Rollenmodells sichtbar machen. Dies beinhaltet eine Funktion zur Auflistung der erzeugten sSoD-Instanzen, eine Funktion zur Rückgabe der mit einer Instanz verknüpften Rollen sowie eine Funktion zur Rückgabe der Kardinalität.

- `sSoDRoleSets`: Liefert die Menge der definierten Instanzen einer sSoD-Relation zurück.
- `sSoDRoleSetRoles`: Liefert die Menge an Rollen einer sSoD-Instanz zurück.
- `sSoDRoleSetCardinality`: Liefert die Kardinalität einer Instanz zurück.

Funktionale Spezifikation für die dynamische SoD-Relation

- **Administrative Funktionen mit *constraints*.** Die Semantik der Erzeugung einer dSoD-Relation stimmt mit der Semantik bei einer sSoD-Relation überein. Während die *constraints* bei sSoD angewandt werden, sobald sich Änderungen an einer *user assignment* ergeben, greifen dSoD-*constraints* typischerweise nur zu dem Zeitpunkt, an dem Rollen für einen Kontext aktiviert werden. Folgende Funktionen werden in diesem Modell spezifiziert:
  - `CreateDSoDSet`: Erzeugt eine Instanz eines dynamischen SoD-*constraint*.
  - `DeleteDSoDSet`: Löscht die Instanz eines dynamischen *constraint*.
  - `AddDSoDRoleMember`: Fügt eine Rolle zu einem bestehenden dSoD-*constraint* hinzu.
  - `DeleteDSoDRoleMember`: Löscht eine Rolle aus einer Instanz eines dSoD-*constraint*.
  - `SetDSoDCardinality`: Ändert die Kardinalität für einen dynamischen *constraint*, was darüber entscheidet, über wie viele der Rollen ein Benutzer im gleichen Kontext verfügen darf.

- **Systemunterstützende Funktionen.** In *constrained* RBAC ohne Hierarchien sollen die Systemfunktionen in gleicher Weise gelten, wie in *core RBAC*. Für die zusätzliche Funktionalität der *constraints* sind an dieser Stelle Funktionen nötig, die dSoD-*constraints* durchsetzen. Das führt dazu, dass die Funktionen `CreateSession` sowie `AddActiveRole` erweitert werden müssen. Im Falle von `CreateSession` muss während des Aufrufs dieser Funktion sichergestellt werden, dass die für die Sitzung errechneten Rollen in keinem Widerspruch zu einem *constraint* stehen. Im Fall von `AddActiveRole` muss ein bestehendes *constraint* erkannt werden, sobald durch das Aktivieren einer Rolle eine Verletzung eines dynamischen SoD-*constraint* verursacht würde. Die Semantik im Falle von *constrained RBAC with hierarchies* entspricht derjenigen von hierarchischem RBAC:
  - `CreateSession`: Erzeugt eine Benutzersitzung und stellt die aktiven Rollen für den gewählten Kontext unter Beachtung dynamischer SoD-Beschränkungen zusammen.
  - `AddActiveRole`: Fügt eine Rolle in die Menge aktiver Rollen für diese Sitzung ein, falls keine dynamischen SoD-Verletzungen hervorgerufen werden.
  - `DropActiveRole`: Entfernt eine Rolle aus der Menge der aktiven Rollen für die Sitzung.

- **Funktionen zur Überarbeitung.** Zur Implementierung von hierarchischem RBAC mit dynamischen *constraints* sind alle Überarbeitungsfunktionen aus *core RBAC* gültig. So wie für die Überarbeitungsfunktionen im statischen Fall sind auch hier zusätzliche Funktionen nötig, um die drei Parameter zu bearbeiten, die durch das 3-Tupel eingeführt wurden (Geschäftsprozess, Rollen, Kardinalität):

- `dSoDRoleSets`: Liefert für eine dSoD-Relation die Menge definierter Instanzen zurück.
- `dSoDRoleSetRoles`: Liefert die Menge an Rollen zurück, für die eine dSoD-Instanz definiert wurde.
- `dSoDRoleSetCardinality`: Liefert die Kardinalität der Instanz zurück.

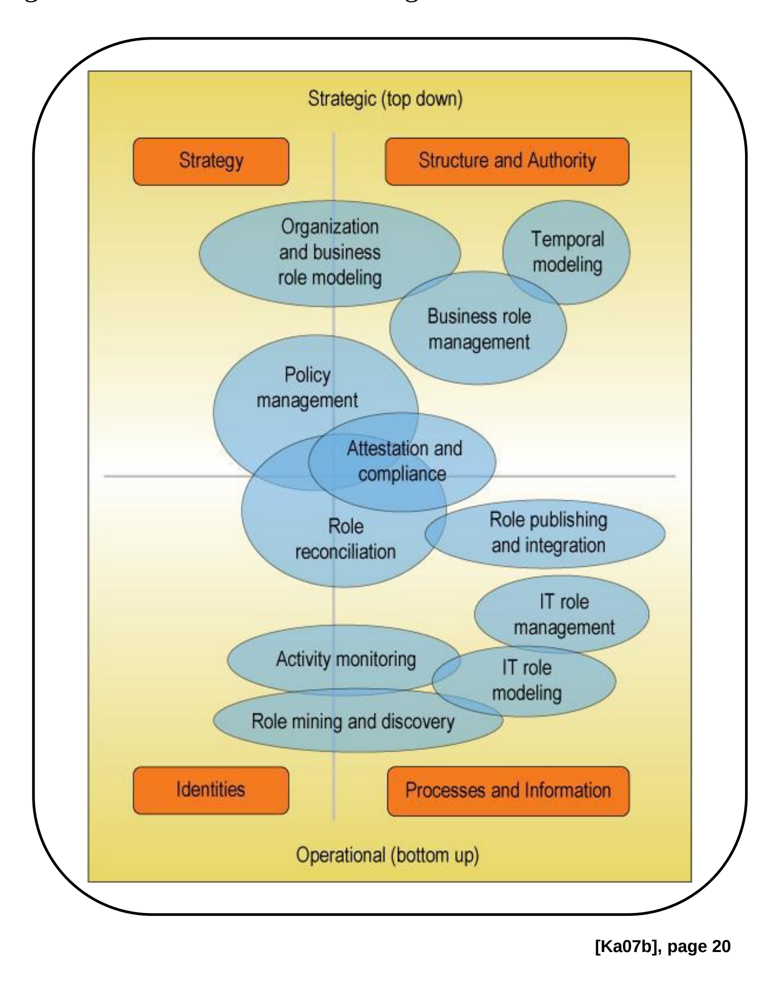
Dies beendet die Einführung in die vier Komponenten des NIST-RBAC-Standards. In den vergangenen beiden Kapiteln 2.1 und 2.2 wurden die Grundlagen für die rollenbasierte Zugriffskontrolle (RBAC) gelegt. Dies bildet die Basis für die Entwicklung eines eigenen Rollenmodells in Kapitel 4. Da ein wesentlicher Teil dieser Arbeit den Bezug zu technischen Umsetzungen von RBAC in Form von kommerziellen Sicherheitsprodukten herstellt, ist es an dieser Stelle nötig, die Grundlagen für diese sogenannten „Rollenmanagementwerkzeuge" zu legen. Im letzten Teilkapitel werden daher nun typische Arbeitsaufgaben und Funktionen dieser Werkzeuge betrachtet.

2.3 Aufgaben von Rollenmanagementwerkzeugen

Die zentralen Aufgaben des Rollenmanagements sind das Definieren, Zuweisen und Pflegen der Beziehungen von Benutzern und Geschäftsrollen, Geschäfts- und Systemrollen sowie Systemrollen und Ressourcen. Insbesondere zwei dieser drei Beziehungen sind dabei problematisch: Wie bereits in Kapitel 1 angesprochen wurde, ist die Beziehung zwischen Benutzern und Geschäftsrollen deshalb schwierig umzusetzen, weil hierzu sowohl die Struktur des Unternehmens, als auch dessen operative Geschäftsprozesse bekannt sein müssen. Bei der Beziehung von Geschäfts- und Systemrollen besteht die Problematik in der großen Zahl an Systemrollen. Diese Komplexität soll werkzeuggestützt reduziert werden. In den letzten Jahren haben sich daher Werkzeuge zur Verwaltung von Rollen auf dem Markt etabliert. Dabei haben sich typische Aufgabenbereiche herauskristallisiert, die in diesem Kapitel grundlegend eingeführt werden, ehe im folgenden Kapitel 3 konkrete Werkzeuge aus dieser Sparte vorgestellt werden. Die in diesem Kapitel vorgestellten Aufgaben werden im Fortgang dieser Arbeit für die Entwicklung eines hybriden Vorgehensmodells in Kapitel 5 ebenso herangezogen, wie für die Entwicklung eines Bewertungskatalogs für Rollenmanagementwerkzeuge in Kapitel 6. Im Folgenden wird zunächst ein Überblick gegeben über fünf verschiedene Kategorien bei der Unternehmensgestaltung und diese anschließend in Bezug gesetzt zu Aufgabenbereichen von Rollenmanagementwerkzeugen. Da die Aufgabenbereiche sehr unterschiedlich sind und innerhalb eines Unternehmens von unterschiedlichem Personal ausgeführt werden, soll diese Unterteilung verdeutlichen, an welcher Stelle im Unternehmen sie im Allgemeinen durchgeführt werden. Hierdurch wird erneut der Bezug hergestellt zu der aus Information 1 bekannten geschäftlichen und technischen Ebene.

Jay R Galbraith beschäftigte sich in [Ga02] mit der Unternehmensgestaltung und hatte dabei insbesondere das Ziel, eine für die unterschiedlichen Ausrichtungen von Unternehmen möglichst effiziente Organisationsstruktur aufzubauen. Dazu führte er in seinem *Star Model* fünf Kategorien ein, die unterschiedliche Aspekte eines Unternehmens betrachten und von denen vier in einen direkten Zusammenhang zu den Aufgaben von Rollenmanagementwerkzeugen gebracht werden. In der als *strategy* bezeichneten Kategorie werden die zentralen Ziele eines Unternehmens definiert, dessen produzierte Güter bzw. Dienstleistungen sowie dessen Philosophie. Die *strategy* gibt somit die Ausrichtung des Unternehmens an und wirkt sich damit unmittelbar auf die spätere Organisationsform aus. Ausgehend von der in der Kategorie *strategy* definierten Ausrichtung, befasst man sich in der zweiten Kategorie *structure and authority* mit der Aufteilung von Verantwortlichkeiten. Hier wird dem Unternehmen somit eine Struktur gegeben und Verantwortlichkeiten definiert und verteilt. *Processes and information* füllt die definierte Struktur nun mit konkreten Geschäftsprozessen und Arbeitsaufgaben. Die Kategorie *rewards* befasst sich mit der Einführung von Boni, um die Ziele eines Unternehmens mit den individuellen Zielen der Angestellten in Einklang zu bringen. Diese Boni sind nicht ausschließlich monetärer Natur, müssen aber in Einklang mit der bislang definierten Unternehemsausrichtung, dessen Struk-

tur und Arbeitsprozessen stehen. Die fünfte Kategorie *people and resources* oder *identities* beschäftigt sich mit der Akquise und Einführung von neuem Personal sowie dessen Rotation und persönlicher Entwicklung im Unternehmen [Ga02, Kapitel 2]. Die Kategorien *strategy* und *structure and authority* geben dem Unternehmen das strukturelle Rückgrad und entsprechen somit der in dieser Arbeit verwendeten Geschäftsebene, wohingegen *people and resources* und *processes and information* eher technische Spezifika betrachten und somit der technischen Ebene anzugliedern sind [Ka07b]. Die Kategorie *reward* steht in keinem konkreten Zusammenhang zum Rollenmanagement und wird deshalb im Folgenden nicht weiter betrachtet.



[Ka07b], page 20

Information 11: Basics – Role Management Capabilities

In Information 11 werden die Aufgaben von Rollenmanagementwerkzeugen grafisch dargestellt. Dabei sind die Aufgaben, die eher im Bereich der Geschäftsebene angesiedelt sind, in der oberen Hälfte und die Aufgaben der technischen Eben in der unteren Hälfte angeordnet. Zusätzlich teilt die vertikale Hilfslinie diese Aufgaben in die vier angesprochenen Kategorien. Die elf verschiedenen Aufgaben des Rollenmanagements stammen aus [Ka07b] und werden hier von der Geschäftsebene aus nach unten erklärt.

- Um die Funktionen eines einzelnen Angestellten korrekt zu erfassen, ist es nötig, ihn im Zusammenhang zur gesamten Unternehmensstruktur und dem Geschäftsziel des Unternehmens zu betrachten. Auch die Abgrenzungen an den Schnittstellen zu anderen Verantwortlichkeiten sind hierfür wichtig. Eine Geschäftsrolle stellt eine Repräsentation dieser angesprochenen Funktionen dar und so müssen sich in ihr die Charakteristika des Unternehmens widerspiegeln. Insbesondere ist es wichtig, die Beziehungen dieser Geschäftsrolle zu anderen zu erfassen. Das *organization and business role modeling* stellt Mechanismen zur Verfügung, die es ermöglichen, diese Unternehmenseigenschaften

festzuhalten und den Zusammenhang zwischen den Geschäftsrollen herzustellen [Vaau07b].

- Ein Unternehmen ist ständigen Veränderungen ausgesetzt. Dies umfasst beispielsweise die Einstellung neuer Mitarbeiter, den Zukauf weiterer Firmen oder etwa Änderungen der Befugnisse von Mitarbeitern. Für ein Unternehmen ist es essentiell, sowohl Änderungen zurückverfolgen zu können, als auch Auswirkungen von Änderungen in der Zukunft messbar machen zu können. Rollenmanagementwerkzeuge ermöglichen es im *temporal modeling and state management*, Änderungen zurückverfolgbar oder vorhersagbar zu machen. So sollte es zurückverfolgbar sein können, über welche Rollen ein Angestellter zu einem bestimmten Zeitpunkt verfügt hat und von wem diese Rollen erteilt wurden. Die Vorhersagbarkeit hingegen dient der Planung von Änderungen an Rollenzuweisungen oder Rollenstrukturen und misst deren Auswirkungen auf das Rollenmodell.

- Aus der Geschäftssicht repräsentiert eine Geschäftsrolle eine Menge an Aufgaben, die ein Angestellter in seiner Position im Unternehmen ausführt. Je nach Aufgabe kann der Umfang einer Geschäftsrolle sehr limitiert, aber auch sehr groß sein. Das Ziel eines guten Rollenkonzepts ist es, den Umfang einer Rolle so zu definieren, dass sie in möglichst vielen Kontexten wiederverwendet werden kann. Ein Rollenmanagementwerkzeug ermöglicht es im Rahmen des *business role management*, Rollen zu definieren, sie zu verwalten und zu analysieren, um die Granularität des Umfangs anzupassen.

- Primär besteht eine Rolle aus Attributen, wie eine Liste von Benutzern im Fall von Geschäftsrollen, oder eine Liste von Zugriffsrechten bei Systemrollen. Darüber hinaus kann sie über Einschränkungen (engl. *constraints*) verfügen, die in Form einer Policy realisiert werden. Über diese Einschränkungen ist es etwa möglich, ihren Gültigkeitsbereich zeitlich oder örtlich zu beschränken. *Separation of duty* (SoD) wird ebenfalls auf diese Weise realisiert. Eine Rollenmanagementlösung stellt im Aufgabenbereich *policy definition and management* Möglichkeiten bereit, um Policys zu erstellen und sie zu verwalten. Auch gibt es Werkzeuge, die darüber hinaus das automatisierte Herausarbeiten von Policys und das Zuweisen zu Rollen anbieten.

- Für das Personal, das sich um die Zuweisungen der Rollen kümmert, beginnen Überwachungstätigkeiten, sobald die Zuweisung von Rollen an Benutzer erfolgt ist. Aufgaben für diesen Bereich versteht man unter *attestation and compliance collection and reporting*. Dies umfasst eine regelmäßige Überprüfung, die feststellt, ob die erteilten Zuweisungen noch wirksam sind. Auch muss sichergestellt werden können, dass mit Verstößen gegen Geschäftsregeln oder -fristen (engl. *compliance issues*) effektiv umgegangen wird. Zusätzlich dazu sollte bei Verstößen gegen das SoD-Prinzip das verantwortliche Personal automatisch gewarnt und entsprechende Schritte eingeleitet werden.

- Zwar bieten Rollenmanagementwerkzeuge das Erzeugen abstrakter Geschäfts- und Systemrollen an, es ist aber ebenso wichtig, die Beziehung zwischen Geschäftsrollen und Benutzern sowie Systemrollen und Zugriffsrechten zu betrachten. Aufgrund der Vielzahl unterschiedlicher Rollen ist es wichtig, dass ein Rollenmanagementwerkzeug im Aufgabenbereich *role reconciliation* unterschiedliche Filter und Sichtweisen anbietet. Entscheidend ist dabei, sehen zu können, über welche Rolle oder Rollen ein einzelner Benutzer verfügt, von wem und wann die Zuteilungen erfolgt sind und zu welchem Zeitpunkt sie im Zuge einer routinemäßigen Verwaltungsaufgabe überprüft werden sollte. Dies sollte sowohl für Geschäfts- als auch für Systemrollen möglich sein.

- Da Rollen und Policys in Rollenmanagementwerkzeugen aus verschiedenen Quellen akquiriert und analysiert werden, haben diese Werkzeuge einen Überblick über das ganze Unternehmen. Daher macht es Sinn, sie auch als autorative Quelle für diese Informa-

tionen zu verwenden. So könnten Eigenschaften von Rollen – wie etwa Einschränkungen oder Mitgliedschaften – interessant sein für Systeme, die selbst keinen Zugang zu diesen Informationen haben, oder mit diesen Informationen nicht versorgt werden. Im Falle von Policys würde die Rollenmanagementlösung dadurch zu einem Policy Information Point (PIP) für die gesamte Umgebung. Eine bestehende Zugriffskontrollarchitektur könnte direkt mit einem PIP kommunizieren und würde somit von einer Komponente, die Policy-Informationen zentral kapselt und verwaltet, direkt profitieren. Diese Information über einen Dienstzugangspunkt mit Standardschnittstellen anzubieten, wäre besonders beim Umstieg auf eine Rolleninfrastruktur nützlich, wo oftmals Altsysteme (engl. *legacy systems*) vorhanden sind. Die Prinzipien *role publishing* sowie *policy publishing* steuern somit einen wertvollen Beitrag hin zu einer wohldurchdachten Infrastruktur bei. Neben dem passiven Anbieten von Rollen- und Policy-Informationen ist der Arbeitsbereich *role integration* aktiv an einer Integration mit anderen Lösungen beteiligt. Dabei werden Änderungen an Rollen oder Policys an andere Zugriffskontrollsysteme weitergereicht. Die Anforderungen an die angebotenen Schnittstellen sind in diesem Fall wesentlich höher.

- So wie Geschäftsrollen benötigen auch Systemrollen einen Mechanismus zur Definition, Verwaltung und Suche, um erfolgreich eingesetzt werden zu können. Dies wird unter dem Begriff *IT role management* verstanden.

- Das aktive Überwachen der Zugriffe von Benutzern auf Ressourcen ist Bestandteil des *activity monitoring and correlation* und ist eine weitere Hilfe beim Erstellen von Rollen, weil so Zugriffsmuster entdeckt werden können und die Granularität der Rollen passgenau an das Zugriffsverhalten der Benutzer angeglichen werden kann. Werden Ressourcen über einen gewissen Zeitraum hinweg überwacht, können gehäuft auftretende Zugriffsversuche aufgezeichnet und dadurch unangemessene Zugriffsberechtigung von Benutzern entdeckt werden.

- Auf technischer Ebene ist eines der Hauptziele des Rollenmanagements, die Zugriffskontrolle auf Ressourcen so einfach wie möglich zu gestalten. Es wird daher versucht, möglichst allgemeine Zugriffsrechte zu konzipieren, die bei Zuweisungen zu Geschäftsrollen möglichst oft wiederverwendet werden können. Das Rollenmanagement sollte im *IT role modelling* Möglichkeiten zur Verfügung stellen, diese Zugriffsrechte in Systemrollen zu kapseln. Darüber hinaus sollte eine Zuordnung von Geschäfts- und Systemrollen möglich sein.

- Auf unterster technischer Abstraktionsebene ist es nötig, Benutzer- und Autorisierungsinformationen aus einer Menge unterschiedlicher Systeme beziehen zu können. Diese Aufgabe wird als *role discovery* bezeichnet und stellt im Allgemeinen den ersten Schritt bei der Umstellung einer Struktur auf Rollen dar. Im laufenden Betrieb befasst sie sich in einem iterativen Prozess mit der schrittweisen Verfeinerung der herausgearbeiteten Rollen und den Beziehungen zwischen ihnen, was unter dem Begriff *role mining and discovery* zusammengefasst wird.

Diese Komponenten spiegeln sich ganz oder zum Teil in den Rollenmanagementwerkzeugen wider. Zusätzlich zu diesen elf Komponenten bildet die Konformität zum RBAC-Standard [FS+01] die Grundlage für eine rollenbasierte Infrastruktur. Hier werden die Beziehungen im Umgang mit Benutzern, Rollen und Ressourcen beschrieben sowie Hierarchien in Rollenbeziehungen und SoD spezifiziert [Ka07b].

In diesem Kapitel sind die Grundlagen für die Entwicklung von Rollenmodellen für die Zugriffskontrolle gelegt worden. Dabei wurde zunächst auf das Prinzip der rollenbasierten Zugriffskontrolle im Bezug zur Subjekt/Objekt-Relation eingegangen und im Anschluss daran der NIST-RBAC-Standard mit den darin enthaltenen vier Rollenmodellen vorgestellt. Zum Ab-

schluss wurden typischen Aufgabenbereiche von Rollenmanagementwerkzeugen aufgeführt und in Bezug gesetzt zu den zwei unterschiedlichen Geschäftsebenen: Auf der geschäftsnahen Ebene hat man es in erster Linie mit organisatorischen Aspekten des Unternehmens, den Geschäftszielen oder Geschäftsprozessen zu tun, wohingegen auf technischer Ebene die Endsysteme sowie die Verwaltung und Pflege von technischen Berechtigungen bzw. Policys im Vordergrund stehen. Im folgenden Kapitel über den Stand der Technik werden nun, aufbauend auf diesen Grundlagen, aktuelle Forschungsergebnisse und Betätigungen zusammengetragen, die im unmittelbaren Zusammenhang zur Entwicklung eines Rollen- sowie Vorgehensmodells für den Unternehmenskontext stehen sowie eine detaillierte Einführung in zwei ausgewählte Rollenmanagementwerkzeuge gegeben.

3 STAND DER TECHNIK

In diesem Kapitel wird ein Überblick über den aktuellen Stand bei der rollenbasierten Zugriffskontrolle innerhalb von Wissenschaft und Technik gegeben. Das Kapitel ist dabei folgendermaßen gegliedert: Kapitel 3.1 betrachtet zunächst aktuelle Modelle zur rollenbasierten Zugriffskontrolle, die speziell für den Einsatz in Unternehmen konzipiert wurden und unterschiedliche Aspekte beleuchten. Dies stellt die Verknüpfung zum vorangegangen Kapitel her, in dem die Grundlagen für diese spezialisierte Betrachtungsweise gelegt wurden. Zunächst wird dabei ein Modell zum automatisierten Entwickeln von Rollen vorgestellt, an das sich die Betrachtung eines speziellen Rollenmodells für den Unternehmenskontext anschließt. Abschließend wird auf den Lebenszyklus von Rollen im Wirkbetrieb eingegangen. Mit dem Begriff „Lebenszyklus" wird ein Vorgehensmodell bezeichnet, welches an den klassischen Software-Entwicklungszyklus angelehnt ist, diesen auf die Entwicklung von Rollen überträgt und dabei administrative Aufgaben im Wirkbetrieb eines instanziierten Modells explizit beachtet. Kapitel 3.1 zeigt somit aktuelle Forschungsergebnisse im Bereich RBAC aus der Wissenschaft auf, ehe in den Kapiteln 3.2 und 3.3 auf kommerzielle Implementierungen eingegangen wird. Bei diesen beiden Rollenmanagementwerkzeugen handelt es sich um den Identity Manager der Firma Omada und den Role Manager der Firma Sun. Die beiden Kapitel sind in analoger Weise so aufgebaut, dass zunächst ein Gesamtbild der Architekturen und Einzelkomponenten gegeben wird. Daran schließt sich eine genauere Betrachtung jedes dieser Komponenten an, wobei auch darauf geachtet wird, eine Verbindung zu den typischen Aufgaben von Rollenmanagementwerkzeugen aus Kapitel 2.3 herzustellen. Das Ziel der beider Kapitel 3.2 und 3.3 ist es, einen wertungsfreien Überblick über den aktuellen Stand in der Technik zu geben.

3.1 Modelle für die rollenbasierte Zugriffskontrolle

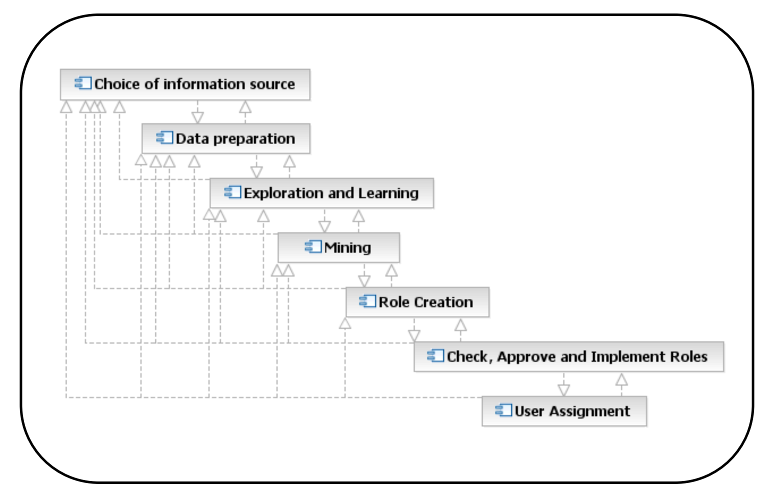
3.1.1 Modell zur Entwicklung von Rollen

Ein Ansatz, der mittlerweile sowohl in der Forschung, als auch in der Technik an Beachtung gewinnt, ist das *role mining*. Dieser Kunstbegriff bezeichnet das Anwenden von Algorithmen zur Datengewinnung (engl. *data mining*), um zu geeigneten Repräsentanten von Rollen zu gelangen. Dieses Kapitel bezieht sich auf [KS+03], in dem der *role mining*-Ansatz näher beschrieben wird. Dabei wird in diesem Kapitel zunächst auf zwei allgemeine *data mining*-Techniken eingegangen und diese anschließend auf das *role mining* übertragen. Beim *role mining* wird dann zunächst auf Aspekte bezüglich des Datenbestands eingegangen, die beachtet werden müssen und anschließend auf den *role mining*-Prozess selbst. Um diesen Ansatz einordnen zu können, werden zum Abschluss Erfahrungswerte genannt, die aus dem praktischen Einsatz von *role mining* stammen.

Will man Daten aufbereiten, um daraus Rollen ableiten zu können, muss dieser teilweise sehr umfangreiche Gesamtdatenbestand zunächst auf einen der individuellen Situation angemessenen Ausschnitt beschränkt werden. Weil die Daten aus unterschiedlichen Quellen stammen können, ist es ist somit zunächst nötig, eine Vereinigungsmenge aus diesen Daten zu erstellen und diese anschließend einzugrenzen. Die *data mining*-Techniken, die als Basis des *role mining* eingesetzt werden, lauten Assoziation (engl. *association*) und Clusterbildung (engl. *clustering*) [KS+03]. Mit Hilfe der Assoziationsalgorithmen ist es möglich, Muster in den Datenbeständen zu erkennen, wie etwa, welche Elemente oftmals in Kombination auftreten. Zusätzlich zu den hieraus gebildeten Mengen können Regeln abgeleitet werden, die Aussagen über die Wahrscheinlichkeit des gemeinsamen Auftretens machen. Die zweite Technik der Clusterbildung operiert nun auf diesen Objekten und versucht, Elemente zu gruppieren, die über die gleichen Attribute verfügen. Durch ein iteratives Anwenden dieser Algorithmen mit unterschiedlichen Datensätzen und variierenden Parameterwerten wird der Ergebnisdatensatz schrittweise verfeinert.

Die Information, die zur Definition von Rollen benötigt wird findet sich in verteilten Datenbanken und da Rollen Objekte darstellen, die gemeinsame Zugriffsmuster subsumieren, liegt die Idee nahe, den gerade erwähnten Mechanismus auf Zugriffskontrolldaten zu übertragen. Der *role mining*-Ansatz wendet diesen Mechanismus an, um aus den vorliegenden Daten Rollen abzuleiten. Dabei werden zwei unterschiedliche Ausprägungen des Begriffs „Rolle" unterschieden, die *organizational role* und die *functional role*. Da im weiteren Verlauf dieser Arbeit ebenfalls zwischen zwei Rollentypen unterschieden wird, sei hier erwähnt, dass die beiden Unterscheidungen nicht gleichgesetzt werden dürfen, obwohl sie durchaus gewisse Gemeinsamkeiten haben. Beide Unterscheidungen definieren eine Rolle einerseits als Behälter für geschäftsnahe Funktionen (engl. *organizational role*) und andererseits für Zugriffsrechte (engl. *functional role*). Die Unterschiede zu den beiden zentralen Rollentypen dieser Arbeit werden im Fortgang dieses Kapitels noch genauer betrachtet. *Functional roles* verfügen über eine Menge an Attributen, die allen Benutzern zuteil werden, die in diese Rolle eingetragen sind. Der Anwendung der *role mining*-Algorithmen geht eine Auswahl der Benutzer und Endsysteme voraus, weil hierdurch der Datenbestand auf einen geeigneten Ausschnitt eingeschränkt wird. Für das Entwickeln von Rollen sind die Benutzerkonten selbst mit einzubeziehen sowie deren Attribute und alle technischen Endsysteme, auf die der Zugriff in Form von Rollen gesteuert werden soll. Diese Attribute, zusammen mit den spezifischen Attributen eines Benutzers wie etwa seine Gruppenmitgliedschaften oder systemspezifische Eigenschaften, stellen demnach den Datenbestand des *role mining* dar.

Das Ergebnis der Anwendung der *role mining*-Algorithmen ist die Spezifikation der eben erwähnten Rollentypen. Sie sind folgendermaßen charakterisiert: Die geschäftsnahe Rolle beschreibt den Benutzer im Unternehmen. Laut [KS+03] verfügt ein Benutzer über genau eine solche Rolle und durch sie erwirbt er die Benutzerkonten und die Zugriffsrechte in den Endsystemen. Die Rolle subsumiert sämtliche globale und systemspezifische Informationen, über die ein Benutzer verfügen muss, wenn er in der Rolle tätig wird. Sie ist verknüpft mit denjenigen Komponenten in den Endsystemen, die dort zur Autorisierung eingesetzt werden. Diese können je nach System Benutzergruppen, oder auch nur einzelne Benutzerkonten sein. *Functional roles* dagegen verkörpern weder globale, noch systemspezifische Attribute, sondern stehen für Funktionen innerhalb von Geschäftsprozessen oder der Organisationseinheit, in der sie sich befindet, wobei auch dieser Rollentyp mit Endsystemen verknüpft sein kann. Diese Rollen repräsentieren demnach Funktionen, die über das normale Maß an Zugriffsrechten hinausgehen. An dieser Stelle liegt ein grundlegender Unterschied dieser Rollentypen zu den Rollen aus Kapitel 4, bei denen die explizite Trennung von geschäftlichen und technischen Sichtweisen im Vordergrund steht.



Adapted from [KS+03], Chapter 5

Information 12: State of the Art – The Role Mining Process

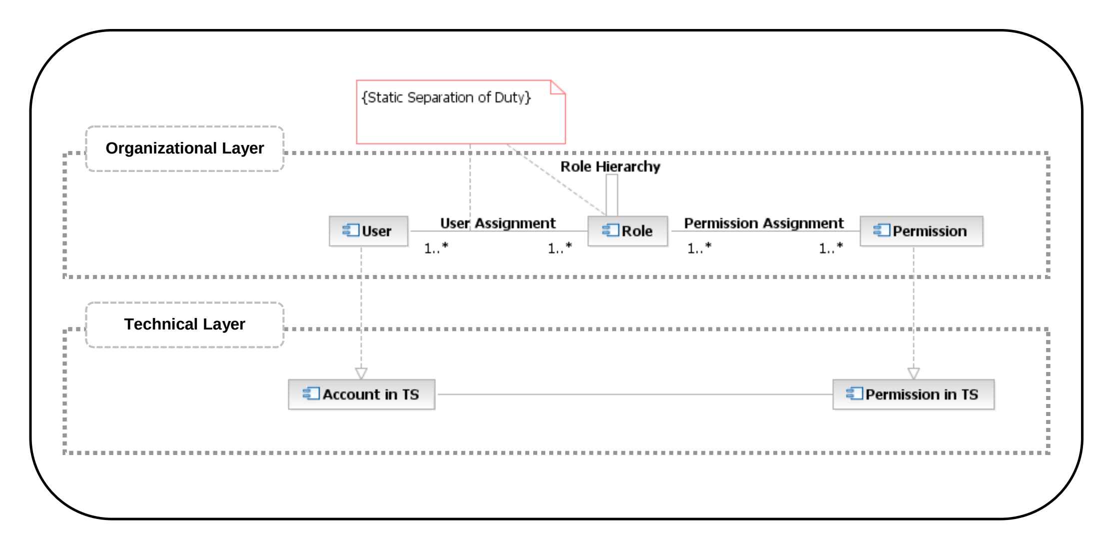
In Information 12 ist der gesamte *role mining*-Prozess dargestellt. Er ist aufgeteilt in sieben diskrete Prozessschritte, die im Folgenden erläutert werden. Man erkennt anhand der Abbildung, dass es sich um einen Prozess mit Rückkopplungen handelt, wobei in jeder Stufe auf jeden der davorliegenden Prozessschritte zurückgesprungen werden kann. Auf die Auswahl einer geeigneten Datenbasis ist bereits eingegangen worden. Das Entscheidende hierbei ist, den gesamten zur Verfügung stehenden Datenbestand auf einen geeigneten Ausschnitt einzugrenzen, um daraus hinreichend geeignete Rollen herauskristallisieren zu können. Dies betrifft sowohl die Benutzerauswahl, als auch die Attributauswahl für eine semantisch hinreichende Untermenge. An dieser Stelle sei darauf hingewiesen, dass die Datenbasis wenig frequentierten Änderungen unterlegen sein sollte, da häufige Änderungen der Datenbasis häufige Anpassungen des Rollenmodells nach sich ziehen [KS+03]. Da bei der Datenauswahl inkorrekte Daten erkannt werden, ist es als Nebeneffekt an dieser Stelle möglich, Inkonsistenzen des Datenbestands zu erkennen und zu beheben, die bei verteilten Daten vorhanden sein können, wie [KS+03] belegt. Nachdem die Auswahl und Sondierung des Datenbestands abgeschlossen ist, folgt in Stufe zwei die Vorbereitung der Daten (engl. *data preparation*). Nachdem ein als passend erachteter Ausschnitt des Datenbestands gewählt und eventuell anfallende Korrekturen bewältigt sind, müssen diese Daten in ein Format transformiert werden, welches von der zum *role mining* verwendeten Software lesbar ist. Die Autoren aus [KS+03] verwenden hierfür den IBM Intelligent Miner for Data und im Speziellen die Funktion *demographic clustering*. Für ein vertiefendes Studium von *clustering*-Algorithmen sei verwiesen auf [GR02]. In der dritten Phase liegt der Fokus auf einer Überprüfung der Datenauswahl, ehe sie den bereits erwähnten Prozessen Assoziation und Clusterbildung übergeben wird. Dies ist der Inhalt der nächsten Phase. Ziel bei dieser Überprüfung ist es, ein Gefühl für die Daten sowie das zu erwartende Rollenschema zu bekommen (engl. *exploration and learning*). Initial kann dabei ganz ohne eine Attributauswahl begonnen werden, um aus dem ersten Analyselauf ein Verständnis für die Attribute zu bekommen, die miteinbezogen werden sollen und diese danach ausgewählt werden. Im Anschluss an die Attributauswahl kann der Fokus auf die Attributwerte gelegt werden, da diese zur Clusterbildung analysiert werden. Als Artefakte dieser Phase entstehen somit Attributmengen, die organisatorische und funktionale Aufgaben repräsentieren sowie davon unabhängige Eingabeparameter für die Algorithmen selbst. Nachdem diese Phase abgeschlossen ist und durch schrittweises Verfeinern ein geeigneter Datenstamm für die Rollenbildung definiert wurde, erfolgt die eigentliche Anwendung der Algorithmen. Auch in der Phase der Rollenerzeugung (engl. *role creation*) kann das Ergebnis durch Verfeinern der Parameter schrittweise verbessert werden. Aus den verfeinerten Daten werden dann organisatorische und funktionale Rollen erzeugt sowie deren Verbindungen zu den Endsystemen. Basierend auf den Attributwerten sowie den erzeugten Attributclustern werden die globalen bzw. systemspezifischen Attribute erzeugt, die ihrerseits dann zu Attributen von organisatorischen Rollen werden. Nach dieser Entwurfsphase werden die Ergebnisse abschließend evaluiert. Das entstandene Rollengerüst wird auf Plausibilität und Korrektheit überprüft und im Anschluss daran implementiert (engl. *check, approve and implement resulting roles*). Die Implementierung bezeichnet hierbei den Prozess zum Abbilden der Rollen in eine lauffähige Version. Nach Beendigung der Implementierung des Rollenmodells ist der letzte Schritt in diesem Prozess das Zuweisen von Benutzern zu diesen Rollen (engl. *user assignment*), was laut [KS+03] manuell erledigt werden muss.

Abschließend sollen Erfahrungswerte dieses Ansatzes genannt werden, die Aufschluss über die Effektivität des *role mining*-Ansatzes geben. In [KS+03] wird auf zwei Fallstudien eingegangen, wobei im ersten Fall die Zeit zum Entwickeln von Rollen im Gegensatz zum klassischen Implementieren von Rollen näher betrachtet wird. Der zeitliche Aufwand im Vergleich zum Entwickeln ohne Algorithmenunterstützung konnte um zwei Größenordnungen verringert werden; im Speziellen konnte die Entwicklungszeit von einigen Monaten auf einige Stunden verringert werden. Im zweiten Fall lag der Fokus eher auf den Vorteilen im laufenden Betrieb. Dieser Modellrechnung lag die Voraussetzung zugrunde, dass die Anzahl der Mitarbeiter jährlich um 10 % zunehmen würde und die Anzahl der Rollen im gleichen Zeitraum um 5 %. In diesem Fall konnte eine Kostenreduktion bei der Rollenerzeugung von 60 % und im Betrieb von 50 % erreicht werden.

3.1.2 Rollenmodell für unternehmensweite Zugriffskontrolle (ERBAC)

Untenehmen sehen sich heutzutage ökonomischen und technischen Änderungen ausgesetzt, die eine höhere Flexibilität und eine höhere Effektivität bei Zugriffskontrollarchitekturen fordern. Eine typische Unternehmenslandschaft besteht heutzutage aus einer Menge unterschiedlicher technischer Systeme, wie etwa Betriebssystemen, Datenbanken und Anwendungen. Der Zugriff auf diese Endsysteme wird durch eigene Sicherheitskomponenten oder sogar durch eigenständige Sicherheitsprodukte gewährleistet. Die darin enthaltenen Konzepte unterscheiden sich teilweise jedoch signifikant, oder sind dem Wesen nach unvereinbar. Ein Benutzer, der Zugriff auf diese unterschiedlichen Systeme erhalten soll, muss somit in allen Systemen separat angelegt und gepflegt werden. Aufgrund dieser Unterschiede und der erwähnten Anforderungen im Geschäftsumfeld benötigt ein Unternehmen heutzutage eine durchgängige sowie verständliche Lösung zur unternehmensweiten Zugriffskontrolle.

In diesem Kapitel wird ein Rollenmodell vorgestellt, das sich insbesondere an den Anforderungen von Unternehmen orientiert. Wie [Ke02] in Kapitel 3.2 darlegt, ist dazu eine Modellerweiterung des NIST-RBAC-Standards nötig. Dieser wurden in Kapitel 2.2 eingeführt. In dem hier vorgestellten Modell wird als Erweiterung eine Unternehmensrolle eingeführt, die sich der Heterogenität in den verteilten Systemen von heute stellt. Das Modell ist in der folgenden Abbildung dargestellt.



Adapted from [Ke02], Figure 5

Information 13: State of the Art – Enterprise RBAC Model

Der Rollenbegriff in diesem Modell subsumiert eine Menge an technischen Zugriffsrechten (engl. *permissions*), die nötig sind, um eine Rolle auszuüben. Benutzer werden dazu diesen Rollen zugeordnet, um die darin definierten technischen Zugriffsrechte zu erhalten. Der Unterschied zwischen dem Enterprise RBAC Modell (ERBAC) und dem zugrundeliegenden NIST-RBAC-Standardrahmenwerk liegt neben dem Rollenbegriff im Modellelement Sitzung (engl. *session*), welches in ERBAC nicht vorhanden ist. Unternehmensweite Rollen, wie sie hier verwendet werden, abstrahieren von den technischen Endsystemen und verlieren somit den direkten Bezug zu diesen. In ERBAC wird der Kontext, der ja gerade die Sitzung symbolisiert, implizit durch die Verknüpfung eines Benutzers mit einem oder mehreren technischen Benutzerkonten ausgedrückt, statt ihn explizit zu definieren. Rollen, ihrerseits bestehend aus einer Menge an Benutzern, haben daher keinen direkten Bezug zu einem Kontext. Wie Information 13 zeigt, werden die Berechtigungen, die ein Benutzer durch die Zuweisung zu Rollen erhält, zum entsprechenden Endsystem (engl. *technical system*, TS) durchgereicht, was zu der Erzeugung eines Benutzerkontos (engl. *account*) in diesem System führt. Eine Berechtigung

steht in diesem Modell für eine „Operation" aus dem RBAC Standard und kann jede Art der Autorisierung darstellen, die in dem entsprechenden Endsystem formuliert werden kann. In Anlehnung an das *hierarchical* RBAC-Modell wird auch in diesem Rollenmodell eine Rollenhierarchie eingeführt, was durch einen azyklischen Graph dargestellt wird. Ziel der Hierarchien ist einerseits das Abbilden von Organisationsstrukturen auf das Rollenmodell und andererseits das Vererben von Rechten, so dass Rollen alle Berechtigungen an diejenigen Rollen weitergeben, die im Hierarchiebaum unter ihnen angesiedelt sind. Der RBAC Standard definiert im *constrained* RBAC-Modell Beschränkungen (engl. *constraints*). Dieser Mechanismus wird in Übereinstimmung mit dem Standardmodell in dem hier vorgestellten Modell dazu verwendet, um den statischen wechselseitigen Ausschluss von Rechten (engl. *static separation of duties*, sSoD) durchzusetzen. Realisiert wird sSoD in Form von Regeln, die eine Relation zwischen sich wechselseitig ausschließende Rollen herstellen und ausgewertet werden, sobald sich an einer Rolleneinteilung etwas ändert. Das statische SoD gewährleistet eine konsistente Rollenstruktur und verhindert Einteilungen, die allgemein nicht erlaubt sind. Da das ERBAC Modell keine Sitzungen definiert, ist es hier nicht möglich, dynamisches SoD direkt zu formulieren. Stattdessen muss es sich darauf verlassen, dass entsprechende Mechanismen in den Endsystemen existieren. Die Autoren des ERBAC-Modells geben die Aufgabentrennung durch die Verwendung dedizierter Benutzerkonten als Möglichkeit, dynamisches SoD zu modellieren. Ein Benutzer legt somit seinen Kontext dadurch selbst fest, dass er jeweils unterschiedliche Benutzerkonten verwendet [Ke02].

3.1.3 Das erweiterte ERBAC-Modell

Wie aus [Ke02] hervorgeht, stellt ERBAC ein gutes Modell für verteilte Informationssysteme dar, wie man sie heutzutage typischerweise in Unternehmen vorfindet. Das liegt in erster Linie an der Spezifikation von Unternehmensrollen, die von konkreten Endsystemen abstrahieren. In der praktischen Umsetzung weist es jedoch einige Schwachstellen auf, die im Speziellen zu einem höheren Administrationsaufwand führen. Aus diesem Grund wurden Erweiterungen dazu definiert, die in diesem Teilkapitel vorgestellt werden sollen. Die Erweiterungen beziehen sich auf [Ke02], wo sie anschaulich dargelegt und begründet werden. Diese Betrachtung bildet die Grundlage für die Entwicklung eines Rollenmodells in Kapitel 4.

Der hohe Administrationsaufwand in der praktischen Umsetzung des ERBAC-Modells wird laut [Ke02, Kapitel 4] dadurch hervorgerufen, dass es zu einer großen Anzahl an Rollen führt. Dafür werden zwei Gründe angegeben:

- Es gibt viele Faktoren, die eine Rolle klar definieren und man hat bei der Umsetzung von ERBAC prinzipiell zwei Möglichkeiten, diese Faktoren im Rollenmodell abzubilden. Einerseits können Informationen in Form von Hierarchien dargestellt werden und andererseits können sie durch Attribute in der Rolle selbst festgehalten werden. Eine Rolle ist demnach durch die Menge an Attributen und den dafür definierten Attributwerte eindeutig definiert. Jede Konfiguration dieser Attribute definiert somit eine Rolle eindeutig. Beispiele für Attribute sind etwa die Zugehörigkeit zu einer Organisationseinheit wie etwa einer Abteilung, das Arbeitsprofil der Rolle, den Ort des Arbeitsplatzes oder Ähnliches. Jede Rolle verfügt nun in diesen Attributen über eine eindeutige Wertbelegung. Da eine Rolle genau dieser eindeutigen Wertbelegung ihrer Attribute entspricht, führen Änderungen der Werte somit automatisch zur Entwicklung zusätzlicher Rollen. Um die Anzahl der Attribute pro Rolle zu verringern, lässt sich der Hierarchiemechanismus verwenden: Hierbei werden gleiche Attribute mehrerer Rollen aus diesen extrahiert und eine übergeordnete Rolle dafür definiert. Dadurch erreicht man eine Hierarchiebildung auf der Basis dieses speziellen Attributs. Da es aber in der Regel sehr viele unterschiedliche Hierarchien gibt, würde das Rollenmodell auch sehr viele voneinander unabhängige Rollenhierarchien aufweisen. Beispielsweise hat die Hierarchie der Angestellten einer Firma nicht unmittelbar etwas mit der Organisationsstruktur der Firma zu tun. Das Pflegen von mehreren unabhängigen Hierarchien verursacht eine sehr

hohe Komplexität in der Umsetzung, was bei einem Rollenmodell als sehr kritisch eingestuft werden muss, weil die Verringerung der Komplexität ja gerade eines der Hauptziele von RBAC darstellt. Wie gerade angesprochen wurde, kann diese Information als Alternative zur intensiven Nutzung von Hierarchien als Attribut implizit in den Rollen selbst mitgeführt werden. Dies jedoch führt dazu, dass eine Rolle tendenziell über sehr viele Attribute verfügt. Auch hierbei gilt, dass bei einer Unterscheidung in nur einem der Attributwerte bereits eine neue Rolle definiert werden muss. Daher wächst die Zahl der Rollen im ERBAC-Modell im praktischen Einsatz sehr stark an und zieht eine komplexe Rollenstruktur nach sich.

- Man möchte eine möglichst feingranulare Kontrolle über die Zugriffsrechte in den Endsystemen haben. In typischen Geschäftsanwendungen ist diese feingranulare Kontrolle oftmals deshalb vonnöten, weil die Zugriffsrechte für unterschiedliche Prozesse ebenfalls sehr unterschiedlich sein können. Hierbei werden die Rechte durch unterschiedliche Wertebelegungen in den definierten Attributen ausgedrückt. Beispielsweise haben Bankangestellte unterschiedlich hohe Bewilligungsgrenzen für das Attribut „Kreditrahmen“. Selbst wenn der Wert für das Attribut „Kreditrahmenbeschränkung“ vielleicht der einzige Unterschied in den Zugriffsrechten zweier Bankangestellter ist, manifestiert sich dieser Sachverhalt im ERBAC-Modell in zwei eigenständigen Rollen, weil dies nicht anders zu modellieren ist (vgl. Information 13).

Um diese Probleme zu lösen, wird eine Parametrisierung der Rollen als Erweiterung zum ERBAC-Modell in [Ke02] vorgeschlagen. Dies kommt durch die Verwendung von Attributen und Regeln zum Ausdruck. Dabei können folgende Modellelemente des ERBAC-Modells, das in Information 13 dargestellt ist, parametrisiert werden: Benutzer (engl. *user*), Rolle (engl. *role*), die Benutzer/Rolle-Relation (engl. *user assignment*), die Rolle/Berechtigung-Relation (engl. *permission assignment*) sowie die Rollenhierarchie (engl. *role hierarchy*). Dabei steuern Regeln, welche Schritte zu ergreifen sind, wenn Attribute oder Zuteilungen (engl. *assignments*) geändert werden. In den folgenden Unterkapiteln wird nun auf die vier Erweiterungen eingegangen, die die Anzahl der implementierten Rollen und damit auch den administrativen Aufwand sehr stark verringern. Diese vier Erweiterungen basieren auf Erfahrungswerten aus dem Umgang mit dem ERBAC-Modell und haben sich in der Praxis bereits bewährt, wie [Ke02] belegt.

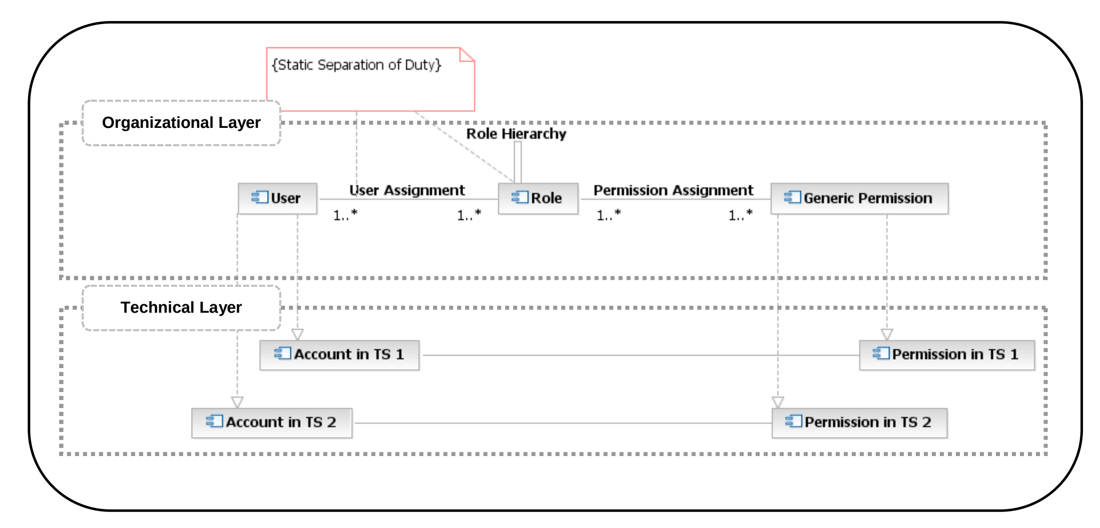
Benutzerattribute

Eine erste Erweiterung des ERBAC-Modells aus Kapitel 3.1.2 stellen die Benutzerattribute dar. Das Modellelement *user* aus Information 13 verfügt bereits über Standard- sowie benutzerspezifische Attribute. Standardattribute sind Eigenschaften, die alle Benutzer besitzen, wie beispielsweise der Name des Benutzers, seine Anrede, oder seine Emailadresse. Diese Informationen können in einem automatisierten Provisionierungsprozess an die Endsysteme weitergereicht werden, wenn dort ein Benutzerkonto für diesen Benutzer angelegt wurde und dieses Benutzerkonto ebenfalls diese Attribute aufweist. Die benutzerspezifischen Attribute wirken sich in einer RBAC-Umgebung auf die Zuweisung von Rollen aus. Hiermit sind Attribute gemeint, wie etwa individuelle Eigenschaften des Jobprofils, die Zugehörigkeit zu einer Abteilung oder Ähnliches. Da die in Attributen vorhandenen Informationen über die Rolleneinteilung entscheidet, bieten sich diese Attribute zur Automatisierung der Rollenzuweisung an.

Generische Rollen

In vielen Unternehmen herrscht eine Koexistenz verschiedener lokaler Endsysteme, wie etwa technisch identischer Endsysteme, die in mehreren Standorten unabhängig voneinander eingesetzt werden und über eine eigene Benutzer- sowie Rechteverwaltung verfügen. Zur Abstraktion von der spezifischen Verwaltung etwa von Berechtigungen werden daher spezielle Rollentypen eingeführt, die auch zur Abstraktion von der systemspezifischen Benutzerverwaltung dienen. Ein Unternehmensbenutzer, der an mehreren Standorten arbeitet, verfügt in der Folge der Koe-

xistenz verschiedener Endsysteme über Benutzerkonten in allen diesen Systemen, die ihrer Definition nach technisch allerdings identisch sind. Dies führt insgesamt betrachtet zu einer Koexistenz von Benutzern und Rollenstrukturen in diesen Endsystemen. Um diese Rollentypen zusammenzufassen, werden generische Rollen definiert, die im Gegensatz zu den im ERBAC-Modell definierten Rollen keine systemspezifischen Berechtigungen zu Endsystemen mehr enthalten, sondern generische Berechtigungen. Diese zeichnen sich dadurch aus, dass sie für eine Menge von technischen Systemen definiert worden sind. Erhält ein Benutzer eine generische Rolle, müssen ein oder mehrere Endsysteme explizit angegeben werden, zu denen der Benutzer Zugang erhalten soll. Im Anschluss an diese Auswahl erhält er dann die technischen Berechtigungen dieser ausgewählten Systeme. Generische Rollen sind in Information 14 dargestellt.



Adapted from [Ke02], Figure 6

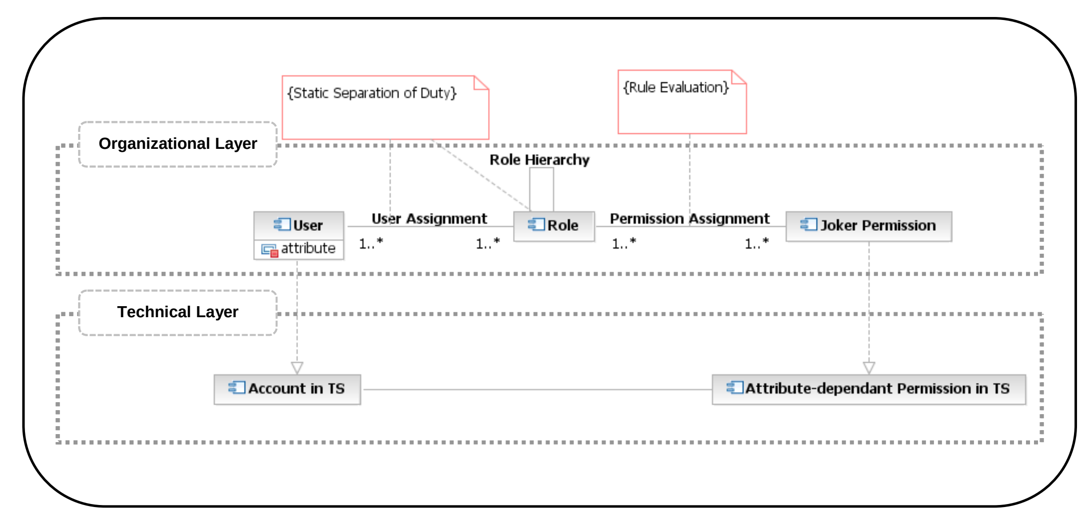
Information 14: State of the Art – Enterprise RBAC Model with Generic Roles

Der hier verwendete Rollenbegriff verkörpert eine generische Rolle, die ihrerseits über eine Menge an generischen Berechtigungen verfügt. Diese generischen Berechtigungen stehen nun für mehrere Endsysteme mit ähnlichen Berechtigungsstrukturen. Wird ein Benutzer in eine Rolle eingeteilt und – wie in der Abbildung dargestellt – zwei Endsysteme ausgewählt, erhält der Benutzer zwei Benutzerkonten (engl. *accounts*) mit den für die Rolle definierten Berechtigungen in diesen Endsystemen [Ke02].

Joker-Berechtigungen

Wie bereits angesprochen wurde, wird eine Rolle durch die Vereinigung aus Informationen verschiedener Strukturen definiert. Zur Veranschaulichung dessen werde etwa definiert, dass die Zugriffsrechte eines Benutzerkontos vom Standort und der Jobfunktion abhängen. Das Ergebnis hiervon ist eine komplexe Rollenstruktur einerseits und eine Vielzahl von Rollen andererseits. Die resultierende Rollenstruktur ist deshalb sehr komplex, weil für jede dieser Eigenschaften ein eigener Vererbungsbaum definiert werden muss und für jede gültige Kombination dieser Eigenschaften eine eigene Rolle definiert werden muss. Im Beispiel würde jedes Benutzerkonto demnach schon über mindestens zwei Rollen verfügen. Um diesen Sachverhalt zu simplifizieren, wird nur eine einzelne Rollenstruktur in Form einer einzelnen Hierarchie aufgebaut und die andere Eigenschaft in Form von Attributen parametrisiert. Sie ist somit lediglich implizit vorhanden. Diese Einschränkung wird als Attribut der Relation Benutzer/Rolle modelliert. Durch eine Joker-Berechtigung ist es nun möglich, lediglich die Attribute zu spezifizieren, deren konkrete Werte allerdings zunächst offen gelassen werden. Bei der Einteilung eines Benutzers in eine Rolle, die über eine Joker-Berechtigung verfügt, wird die effektive Zugriffsberechtigung in Form einer Regel berechnet und somit dynamisch vergeben. Dabei wird der Attributwert aus dem Benutzerobjekt ausgelesen, anstatt ihn manuell vergeben zu müssen. Realisiert wird dies

durch eine Namenskonvention, die vorgibt, dass die Syntax der Attributwerte der Syntax der effektiven Berechtigungen entsprechen muss. Um dies zu verdeutlichen, könnte man sich hier beispielsweise vorstellen, dass alle Standorte durch eine eindeutige Zahl repräsentiert würden. Dann existierten unterschiedliche Gruppen „GroupX“ für den Standort mit der Nummer X. Die Regel der Joker-Berechtigung würde das Benutzerkonto dann entsprechend in die Gruppe X einteilen.

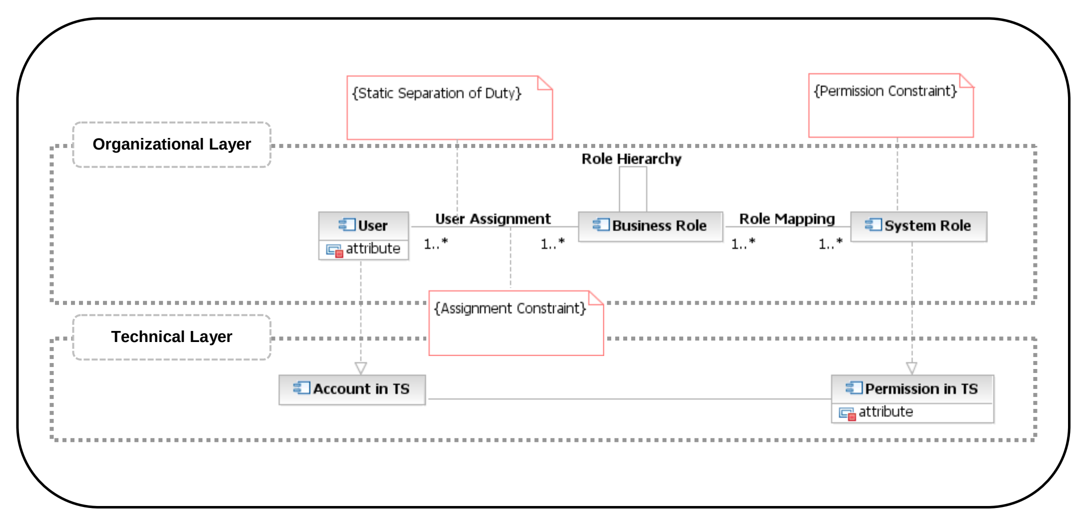


Adapted from [Ke02], Figure 7

Information 15: State of the Art – Enterprise RBAC Model with Joker Permissions

Benutzerspezifische Beschränkungen

Die bisherigen Erweiterungen zum ERBAC-Standard betreffen generische Rollen und die Einteilung zu technischen Berechtigungsgruppen auf der Basis von Attributwerten des Benutzers. Diese letzte Erweiterung beschäftigt sich mit der Anpassung von Berechtigungen auf dynamische Weise, d.h. auf Basis von Benutzerattributen. Innerhalb eines Unternehmens mag es Jobfunktionen geben, die dieselben Aufgaben erfüllen, aber dennoch in einem Parameter unterschiedlich sind. Ein Beispiel soll dies verdeutlichen: In einer Bank gäbe es Angestellte, die Kredite bewilligen dürfen. Je nach Stellung des Mitarbeiters sei dessen Bewilligungsrahmen unterschiedlich hoch. Im ERBAC-Modell aus [Ke02] würde das zu einer Rolle pro Attribut/Wert-Paar führen. Dies kann nun durch benutzerspezifische Beschränkungen parametrisiert werden. Dazu verfügt jeder Benutzer über ein Attribut, welches seine individuelle Grenze festlegt, wie in der folgenden Abbildung verdeutlicht wird.

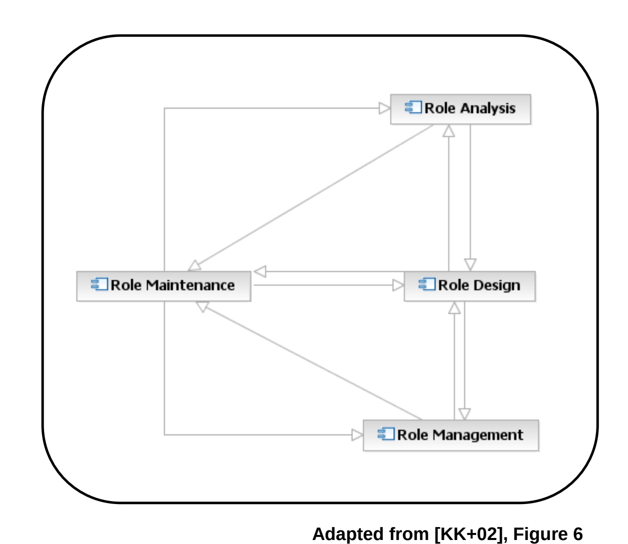


Adapted from [Ke02], Figure 8

Information 16: State of the Art – Enterprise RBAC Model User-specific Constraints

3.1.4 Modell für den Lebenszyklus von Rollen

In Anlehnung an den klassischen Entwicklungszyklus von Software, wie er ist [Ba96, Abbildung 15] beschrieben ist, führt [KK+02] einen Entwicklungszyklus von Rollen ein. Das Besondere an diesem Ansatz ist, dass ein Rollenmodell nicht mit der technischen Realisierung beendet ist, sondern für dessen Verwaltung und Änderungen, die sich aus dessen produktivem Einsatz ergeben, explizite Phasen modelliert. Es ist in diesem Zusammenhang auch vom „Lebenszyklus“ von Rollen die Rede. Es stellt sich nun die Frage, wie relevant die Betrachtung eines Lebenszyklus beim Einsatz von Rollen ist. Dabei verweisen die Autoren auf Erfahrungswerte im Umgang mit dem Lebenszyklus von Rollen durch die Implementierung eines kommerziellen Produktes für den Einsatz in Unternehmen. Wie Information 17 darstellt, bestehet der Lebenszyklus aus vier Phasen, auf die im Folgenden eingegangen wird. Dabei wird beschrieben, welche Aktivitäten in den Phasen enthalten sind.



Adapted from [KK+02], Figure 6

Information 17: State of the Art – The Role Life-Cycle

Rollenanalyse

Die erste Phase befasst sich mit der Analyse von Rollen (engl. *role analysis*) und hat die Aufgabe, geeignete Rollen zu identifizieren, die innerhalb eines Systems verwendet werden. Um die-

ses Ziel zu erreichen, ist es nötig, sowohl explizit vorhandene Strukturen, als auch implizit vorhandenes Wissen zu aggregieren, um eine geeignete Basis für die Entwicklung eines Rollenmodells zu schaffen. Hierzu sind folgende Schritte nötig:

- **Eine Basis für die Geschäftsrollen schaffen.** Das Identifizieren von Geschäftsrollen beginnt bei der Deduktion von geschäftlichen Funktionen aus der Organisation sowie deren Formalisierung und endet bei den Zugriffsrechten auf technischen Systemen. Dies stellt eine rein formale und keine technische Aktivität dar, obwohl an dieser Stelle auch technische Spezifika wie technische Zugriffsrechte betrachtet werden.

- **Hierarchien erkennen.** Nach der Erfassung von geschäftlichen Aspekten werden diese nun zueinander in Bezug gesetzt, um daraus mögliche Hierarchien zu entwickeln. Die entstehende Rollenhierarchie soll dabei möglichst natürlich gewählt sein und den Zustand des Unternehmens möglichst gut repräsentieren. Hierfür kann es auch vonnöten sein, mehrere unterschiedliche Hierarchien aufzubauen.

Die in dieser ersten Phase erzeugten Artefakte sind eher deskriptiver Natur und sollen auf geeignetem Abstraktionsniveau ein Gesamtbild des Unternehmens geben. Diese Artefakte bilden das Unternehmen im Idealfall in sehr anschaulicher Weise ab. Wie [KK+02] erwähnt, sollte in die Analyse hinreichend viel Zeit investiert werden, da ein schlecht erfasstes Gesamtbild zu einem späteren Zeitpunkt mit hoher Wahrscheinlichkeit nachgebessert werden muss, was deutlich mehr Zeit in Anspruch nimmt. Die Analyse bildet die Basis für den Entwurf des Rollenmodells, so dass Nachlässigkeiten in der Analyse zu Rollenmodellen führen, die ihrerseits ebenfalls Schwachstellen aufweisen.

Rollenentwurf

In der Entwurfsphase für Rollen (engl. *role design*) befasst man sich mit dem Entwurf von Rollen auf der Basis der Analyseergebnisse. Ziel dieser Phase ist es, zu einem der Situation entsprechendem Rollenmodell zu gelangen, was robust und praktisch anwendbar ist. Dies beinhaltet die Abbildung von Rollen und den Entwurf des Rollenmodells für die technische Umsetzung. Auf diese beiden Aktivitäten wird im Folgenden näher eingegangen.

- **Abbildung von Rollen.** Um ein Rollenmodell in geeigneter Form abbilden zu können, müssen die organisationsbezogenen Strukturen wie etwa Abteilungsstufen im Rollenmodell abgebildet werden können. Dazu werden die Strukturen in Relation gesetzt zu den Modellen in den Endsystemen. Im Allgemeinen kann die Gesamtstruktur eines Unternehmens als direkter, azyklischer Graph angesehen werden. In diesem Fall kann die Aktivität zur Abbildung von Rollen in die im Folgenden beschriebenen Teile unterteilt werden:
    - Identifikation der Organisationsstrukturen, welche sich im Rollenmodell widerspiegeln sollen. Die dazu notwendigen Artefakte entstammen direkt der Rollenanalysephase.
    - Definition einer injektiven Abbildung, die sowohl die Rollen, als auch die Strukturen auf das Rollenmodell überträgt.
    - Definition einer Benutzer/Rolle-Relation, realisiert auf der Basis von Benutzerattributen. Diese Relation kann als Regel modelliert werden, die auf der Basis von Benutzerattributen über die Rolleneinteilung eines Benutzers entscheidet. Die Benutzerattribute dienen somit dazu, Beschränkungen (engl. *constraints*) zu erfassen. Wenn beispielsweise die Zweigstelle, in der ein Benutzer angestellt ist, eines dieser Attribute ist, kann eine Regel zur automatischen Einteilung in Rollen auf der Basis dieser Attribute formuliert werden. Durch diese Regeln kann die Einteilung in Rollen automatisiert werden, was speziell in der nächste Phase *role management* die Arbeit erleichtert.

- Definition einer Rolle/Berechtigung-Relation, die in analoger Weise zur Benutzer/Rolle-Relation eine Zuweisung von technischen Berechtigungen aufgrund vordefinierter Benutzerattribute vornimmt und die Verwaltung der Rollen (engl. *role management*) vereinfacht.
- Definition eines Algorithmus, mit dessen Hilfe Änderungen an organisatorischen Strukturen auf das Rollenmodell übertragen werden, wobei die bisherigen Zuweisungen (engl. *mappings*) invariant bleiben müssen, um die Konsistenz weiterhin zu gewährleisten. Dieser Schritt ist insbesondere für die Administration der Rollenstruktur im Wirkbetrieb von Nutzen. Im Allgemeinen existieren mehrere, voneinander unabhängige organisatorische Strukturen, in denen sich Änderungen in unterschiedlicher Weise auf das Rollenmodell auswirken. Der Grad an Automatismus und somit der Nutzen des RBAC-Modells, hängt stark davon ab, für wie viele organisatorische Strukturen Algorithmen gefunden werden können, die die Änderungen auf das Modell übertragen. Kann für eine spezielle Änderung kein Algorithmus implementiert werden, kann dieser Bereich auch nicht automatisiert werden.

- **Entwurf des Rollenmodells.** Im Hinblick auf die spätere Administration können beim Entwurf von Rollen Entwurfsmuster unterschieden werden, die hier vorgestellt werden. Für eine eingehende Untersuchung dieser Entwurfsmuster sei an dieser Stelle verwiesen auf [KK+02].

  Es kann Situationen geben, in denen einige Teile des Rollenmodells manuell und andere automatisch verwaltet werden sollen. Dazu kann man mehrere parallele Rollengraphen verwenden, von denen einige Teile der manuellen Verwaltung unterliegen und andere automatisiert verwaltet werden. In diesem Fall existieren mehrere voneinander unabhängige Hierarchien, wobei insbesondere die Schnittstellen zwischen ihnen bzw. die Beziehungen zueinander interessant sind und besonders beachtet werden müssen. Um auch manuelle Aufgaben in der Administration zu ermöglichen ist es wichtig, dass Änderungen in den manuellen Teilen keine der erwähnten Algorithmen ausführen und somit außerhalb des Einflussbereichs der Algorithmen liegen.

  Ein zweites Muster modelliert nur ein einzelnes Rollenmodell. Dieses besteht aus unterschiedlichen Hierarchien und fasst somit semantisch unterschiedliche Begriffen zusammen. Dieser Ansatz vereint demnach unterschiedliche Aspekte in einem einzigen Baum. Dies können beispielsweise organisatorische Aspekte wie etwa der Ort oder die Abteilung sein, kombiniert mit technischen Aspekten wie etwa den Benutzern oder Jobprofilen. Durch die Vereinigung unterschiedlicher Organisationstypen, für die unterschiedliche Arbeitsprozesse notwendig sind, zieht dieser Ansatz für jeden Teilbaum unterschiedliche Automatisierungsprozesse nach sich.

  In einem dritten Ansatz wird der Graph nicht explizit definiert, sondern durch die Anwendung von Algorithmen aus der Graphentheorie dynamisch zur Laufzeit erzeugt. Hierbei werden unterschiedliche Hierarchien durch die Analyse zusammengefasst. Wie [KK+02] aufzeigt, hat es sich in der Praxis bewährt, nur einen Graphen zu verwenden und die logisch unabhängigen Hierarchien in Form von Attributen implizit darzustellen und mit dem Graph in Überdeckung zu bringen. Dies geschieht durch Parametrisierung von Relationen, wie etwa der Benutzer/Rolle-Relation. Dadurch legt man sich auf eine Hierarchie fest und lagert die anderen Hierarchien in die Attribute der im Graphen enthaltenen Knotenobjekte aus. Der Vorteil dieses Ansatzes ist, dass dadurch die strukturelle Komplexität verringert wird, allerdings existieren hierbei auch Nachteile: Durch die implizite Formulierung von Hierarchien in Form von Attributen ist es nicht mehr möglich, Zugriffsrechte auf diejenigen Elemente zu vergeben, die zu Attributen gewordenen sind. Auch gehen durch die Attributierung die Relationen bzw. Hierarchien für diese Elemente verloren.

Rollenverwaltung

Am Ende dieser Phase wurde ein Rollenmodell entworfen, welches über Rollen und Hierarchien verfügt, die der jeweiligen Situation angepasst ist, so dass man sich nun mit der Pflege des Modells befassen kann. Die nächste Phase *role management* befasst sich laut [KK+02] mit administrativen Aufgaben im Wirkbetrieb. Dies sind Routinearbeiten innerhalb des Unternehmens. Die Voraussetzung ist ein robustes Rollenmodell und setzt somit die Analyse- und Entwurfsphase voraus. Die Aufgaben des *role management* umfassen folgende Schritte:

- Die Durchführung von Änderungen am Rollenmodell, was sich mithilfe der bereits definierten Algorithmen in Änderungen der organisatorischen Strukturen auswirkt.
- Das Erzeugen oder Löschen eines Benutzers oder einer Berechtigung.
- Das Zuteilen oder Entziehen von Rollen für Benutzer entsprechend der Benutzer/Rolle-Relation.
- Das Zuteilen oder Entziehen von Berechtigungen für Rollen entsprechend der Rolle/Berechtigung-Relation.
- Das Auftrennen oder Zusammenführen von Rollen selbst, was von besonderer Bedeutung ist. Dies kann entweder dann vonnöten sein, wenn eine komplexe Aufgabe auf unterschiedliche Rollen aufgeteilt werden soll, oder wenn Aufgaben aus mehreren Rollen in einer Rolle zusammengeführt werden sollen.

Rollenwartung

Neben den Änderungen von Rollen und deren Umfang ergeben sich im Laufe der Zeit aber auch Änderungen an den organisatorischen Strukturen selbst, die in der Analyse- und Entwurfsphase ja gerade die Grundlage des zu entwickelnden Rollenmodells bildeten. Auch hier ist der starke Bezug zum klassischen Software-Entwicklungszyklus erkennbar: Die Rollenwartung (engl *role maintenance*) beschäftigt sich nun mit strukturellen Änderungen, wie sie sich etwa beim Zusammenlegen von Abteilungen oder ganzer Firmen ergeben. Im RBAC-Kontext bedeutet das, dass sich Änderungen an den Zuweisungen von den Organisationsstrukturen auf das Rollenmodell ergeben, oder auch an den Relationen Benutzer/Rolle und Rolle/Berechtigung selbst. Der grundlegende Vorteil rollenbasierter Zugriffskontrollarchitekturen zeigt sich hier in besonderem Maße: RBAC-Modelle bieten bei strukturellen Änderungen ein hohes Maß an Flexibilität kombiniert mit der Möglichkeit, sehr schnell auf Änderungen reagieren zu können. Nun setzt man sich in dieser Phase unter Anderem mit der Abbildung organisatorischer Strukturen auf ein mögliches Rollenmodell auseinander, wie es ebenso in der Analysephase geschieht, jedoch müssen Rollen in der Rollenwartungsphase nicht grundlegend neu konzipiert, sondern lediglich an die Änderungen angepasst werden. Dies unterscheidet sich demnach klar von der initialen Analyse. Ferner können diejenigen Teile des existierenden Konzepts, die von den organisatorischen Änderungen nicht betroffen sind, unverändert übernommen werden.

Für eine tiefgreifendere Analyse des Lebenszyklus von Rollen sei verwiesen auf die Beiträge [KK+02] und [SA+04]. Das Kapitel über den Lebenszyklus von Rollen beschießt die Betrachtung aktueller wissenschaftlicher Betätigungen im Bereich der rollenbasierten Zugriffskontrolle für verteilte Informationssysteme. Im Folgenden werden nun zwei kommerzielle Implementierungen von Rollenmanagementwerkzeugen vorgestellt. Beide Lösungen befassen sich mit der rollenbasierten Zugriffskontrolle und sind speziell für den Einsatz in verteilten Informationssystemen konzipiert worden.

3.2 Rollenbasierte Zugriffskontrolle im Omada Identity Manager

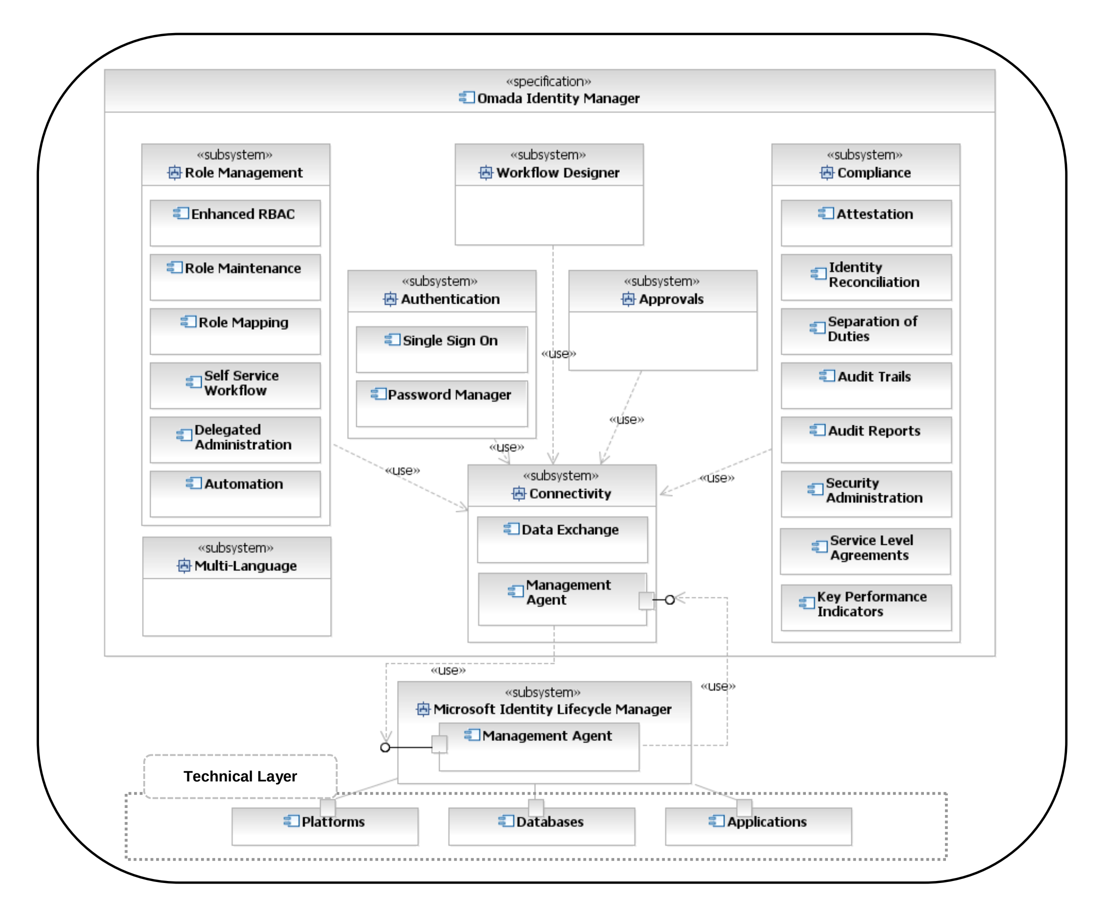
Der Identity Manager der Firma Omada (OIM) ist ein relativ junges Produkt im Rollenmanagement. Die Firma Omada wurde 1999 gegründet und gibt einige erfolgreiche Projektumsetzungen auf Basis des OIM seit dem Jahre 2006 an. OIM basiert auf einer Provisionierungsplattform der Firma Microsoft und dient als Verwaltungswerkzeug für die rollenbasierte Zugriffskontrolle in Umgebungen, die komplett auf Technologien der Firmen Microsoft und

SAP aufbauen. Omada verweist auf über 50 Kunden unterschiedlicher Größe, die die rollenbasierte Zugriffskontrolle (engl. *role-based access control*, RBAC) mithilfe des Omada Identity Manager produktiv einsetzen [Ka07b].

Der Fokus von OIM in der aktuellen Version 6.0 innerhalb des Rollenmanagements liegt auf den Bereichen *role management* und *compliance* und beschäftigt sich somit mit der Verwaltung von Rollen auf der einen Seite und der Sicherstellung, dass Policy-Vorgaben eingehalten werden auf der anderen Seite (vgl. Kapitel 2.3). Dabei protokolliert OIM Änderungen an Rollenzuweisungen und bietet eine Oberfläche zur zentralen Verwaltung von Rollen an. In diesem Kapitel wird zunächst die Architektur des Werkzeugs vorgestellt sowie dessen Datentypen. Es soll hier zunächst ein Überblick über das Produkt gegeben werden, ehe im Anschluss daran eine genaue Betrachtung der beiden Arbeitsbereiche *role management* und *compliance* folgt sowie den jeweils enthaltenen Komponenten. Diese Betrachtung wird in Form eines Prozesses geschildert, um die Anschaulichkeit zu unterstützen. Um ein detailliertes Gesamtbild dieses Werkzeugs zu präsentieren, werden alle Komponenten, die für das Rollenmanagement von Belang sind, objektiv dargestellt. Auf der Basis dieser Produkteinsicht wird in Kapitel 6 dieser Arbeit die Qualität des Omada Identity Manager anhand eines Kriterienkatalogs bewertet.

3.2.1 Architektur und Datentypen des Omada Identity Manager

Der Omada Identity Manager besteht aus mehreren Einzelmodulen bzw. Komponenten, die innerhalb des Gesamtkontextes des Rollenmanagements unterschiedliche Funktionen realisieren. Die Datenbasis, auf die sich die Module stützen und operieren, ist eine OIM-interne Datenbank. Diese synchronisiert sich über die Komponente *connectivity* regelmäßig mit dem Metaverzeichnis des Microsoft Identity Lifecylce Manager (ILM), das als Provisionierungsplattform verwendet wird und die Schnittstelle zu den zugrundeliegenden Endsystemen darstellt. Aus dem Blickwinkel von OIM stellt dieses Metaverzeichnis somit den Gesamtdatenbestand des Unternehmens dar. Der Administrator des Omada Identity Manager kann nun genau spezifizieren, welche Daten aus dem gesamten Unternehmensdatenbestand im Metaverzeichnisses aufgenommen und mit der OIM-internen Datenbank synchronisiert werden sollen. Die Architektur dieses Werkzeugs ist in der folgenden Abbildung dargestellt. Dabei werden die einzelnen Komponenten des OIM in der vorliegenden Version 6.0 in Relation zu seinen Hauptaufgaben gestellt. Um den Bezug zwischen der Architektur und der zentralen Abbildung Information 1 herzustellen, ist die „technische Ebene" (engl. *technical layer*) in der Abbildung ersichtlich.



Information 18: State of the Art – Omada Identity Manager Component Architecture

Information 18 präsentiert die Software-Architektur des Omada Identity Manager. Hier sind die Arbeitsbereiche des OIM mit den darin enthaltenen Komponenten dargestellt und in Bezug gesetzt zur eben angesprochenen „technischen Ebene“ (engl. *technical layer*). Diese Ebene beinhaltet alle im Unternehmen eingesetzten Systeme, aus denen OIM die technischen Informationen, wie etwa Zugriffsberechtigungen, Identitäten der Benutzer oder auch Rollen beziehen kann, falls eines der Endsysteme dieses Konzept bereits unterstützt. OIM kommuniziert mit diesen Systemen über eine Provisionierungsplattform als Middleware somit nur indirekt.

Die Komponente *connectivity* realisiert die Datenkommunikation zu den Endsystemen. Dazu verfügt sie über Managementagenten (engl. *management agents*), über die OIM mit der Provisionierungsplattform kommuniziert. Diese ist dafür verantwortlich, die Daten der angeschlossenen Unternehmenssysteme in einem Metaverzeichnis zu aggregieren und diesen Datenbestand für OIM zugänglich zu machen. Eine zweite Möglichkeit der Interaktion mit den Unternehmenssystemen erfolgt ohne explizite Indirektion durch eine Provisionierungsplattform. Hierbei importiert der Omada Identity Manager aus einer kleinen Auswahl an Datenquellen direkt, oder exportiert dorthin, was in der Abbildung als *data exchange* bezeichnet wird.

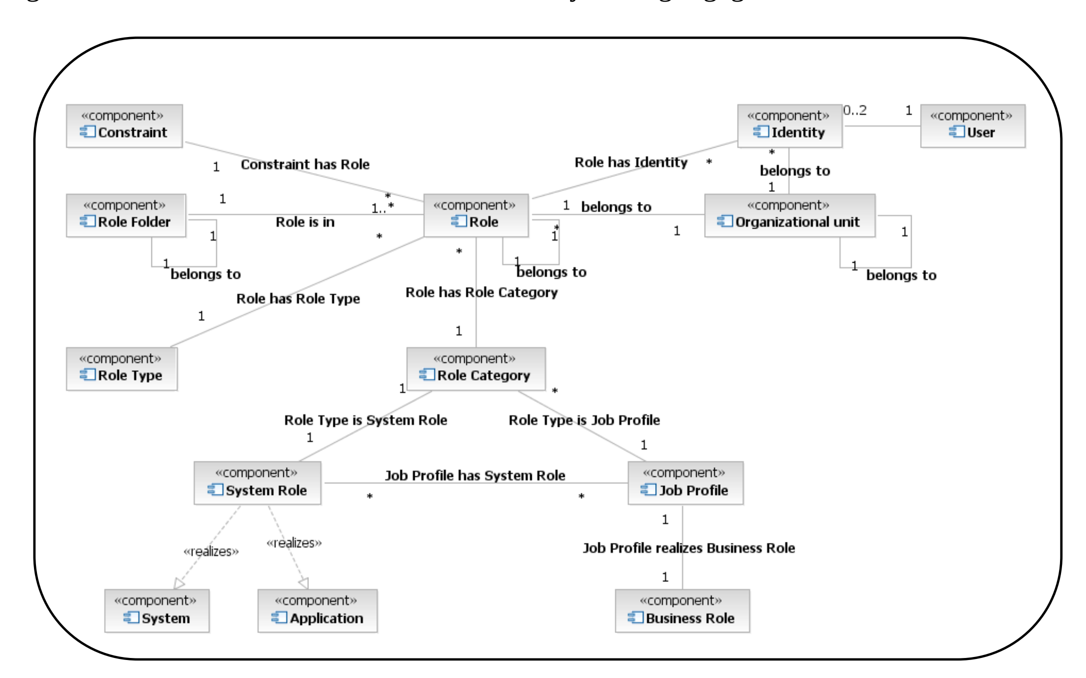
Der Arbeitsbereich *role management* steht für alle Aufgaben, die in OIM im Bereich des Rollenmanagements anfallen. Im Sinne von Information 11 steht dieser Arbeitsbereich für die Aufgaben *organization and business role modeling*, *temporal modeling*, *business role management*, *IT role management* und *IT role modeling*. Die darin enthaltene Komponente *enhanced RBAC* erweitert das zugrunde liegende RBAC-Modell um systemübergreifende Geschäftsrollen, die in der Namensgebung von OIM „Jobprofile“ (engl. *job profiles*) genannt werden. Die Komponente *role maintenance* befasst sich mit der Aufgabe, Änderungen an Rollenzuweisungen vorzunehmen. Dies ist insbesondere bei der Pflege der rollenbasierten Struktur im Wirkbetrieb wichtig. Nachdem Rollen definiert worden sind, können abschließend Benutzer in Relation zu diesen Rollen gesetzt werden. Diese Aufgabe kann je nach Größe und Struktur des Unternehmens sehr

zeitintensiv sein. Die *role mapping*-Komponente bietet für diesen Prozess Unterstützung in Form von Regeln an, die bei Rollenänderungen automatisch ausgewertet werden. Neben der manuellen Zuteilung von Benutzern zu Rollen bietet OIM auch eine automatisierte Zuteilung zu den Rollen an, was mittels Benutzerattributen realisiert ist. OIM bietet die Möglichkeit, die Administration des Gesamtsystems zu dezentralisieren, bzw. administrative Aufgaben von jedem Benutzer selbst initiieren zu lassen. Um dies zu realisieren, ist es in OIM jedem Mitarbeiter möglich, einerseits Änderungen an seinen eigenen Rollenzuweisungen vorzunehmen sowie andererseits für diejenigen Mitarbeiter, für die er verantwortlich ist. Über die Komponente *self service workflow* ist es jedem Mitarbeiter möglich, eigene Rollen zu entfernen, oder die Mitgliedschaft in zusätzlichen Rollen anzufordern. Ferner ist es jedem Benutzer möglich, die in Geschäfts- sowie Systemrollen gekapselten Befugnisse zu einem frei wählbaren Zeitpunkt an eine andere Person zu delegieren, was unter der Komponente *delegated administration* zu verstehen ist. Die letzte Komponente *automation* steht für die OIM-eigenen, automatischen Arbeitsabläufe (engl. *workflows*), die eine Automatisierung im gesamten Zugriffskontrollprozess in vielerlei Hinsicht ermöglichen. Im Zusammenhang mit dem *role management* ist ein automatisches Rückführen von Änderungen des Datenbestandes aus OIM in die Unternehmenssysteme möglich. Dies geschieht durch die Einbeziehung der Provisionierungsplattform. Ein zweites Beispiel für die erwähnten *workflows* sind die Arbeitsbereiche *approvals* und *workflow designer*. Sie stehen für die Bewilligung von Änderungen an der Rollenstruktur sowie für die Modellierung und den Entwurf eigener *workflows*.

Der zweite große Arbeitsbereich *compliance module* befasst sich mit der Einhaltung von Policy-Vorgaben auf unterschiedlichen Stufen des Lebenszyklus einer Rolle. Gemessen an den Aufgaben von Rollenmanagementwerkzeugen aus Information 11 werden hier die Funktionen *policy management, attestation and compliance*, *role reconciliation* und *activity monitoring* realisiert. Durch die Komponente *attestation* wird der Prozess realisiert, der die Zuweisung von Zugriffsrechten überprüft. Dies kann in OIM teilautomatisiert werden, indem dieser Prozess zu regelmäßigen, oder wiederkehrenden Zeitpunkten ausgeführt wird. Durch die Komponente *identity reconciliation*, die einen Abgleich der Identitäten in den Endsystemen vornimmt, wird ermöglicht, dass OIM als zentrale Stelle für die Zugriffskontrolle verwendet werden kann. Hierfür ist es insbesondere notwendig, dass OIM über Änderungen informiert wird, die sich außerhalb der Verwaltung von OIM ergeben. Dies ist zum Beispiel dann der Fall, wenn Zugriffsberechtigungen in einzelnen Unternehmenssystemen selbst verändert werden. Ist dies der Fall, kann das ein Anzeichen dafür sein, dass ein Verantwortlicher aufgrund von Unkenntnis die Zugriffsrechte in diesem Unternehmenssystem vorgenommen hat, statt dies über OIM zu erledigen. Es stellt sich bei Änderungen außerhalb von OIM generell die Frage, ob es sich dabei um geplante Änderungen der Zugriffsrechte handelt, oder um eine Policy-Verletzung. OIM sieht beim Auftreten einer solchen Situation seine eigene Datenbank als privilegiert an und macht die vorgenommenen Änderungen rückgängig, um so die Konsistenz wiederherzustellen. Durch die Komponente Identitätsabgleich (engl. *identity reconciliation*) wird OIM darüber informiert, wenn sich einzelne Daten oder Attribute in den zugrundeliegenden Unternehmenssystemen ändern und kann somit in geeigneter Weise darauf reagieren. Die nächste Komponente zeigt einen deutlichen Vorteil einer rollenbasierten Zugriffskontrolle auf: Die einfache Spezifikation von wechselseitigem Ausschluss. Ein wesentlicher Vorteil einer rollenbasierten Zugriffskontrollarchitektur ist, wechselseitigen Ausschluss von Rechten in konsistenter Form realisieren zu können. Dies wird in OIM in der Komponente *separation of duties* dadurch ermöglicht, dass sich Rollen definieren lassen, die sich gegenseitig ausschließen. Zur Überwachung, ob Policy-Vorgaben eingehalten werden, bietet der Omada Identity Manager durch die Komponenten *audit trail* und *audit report* die Möglichkeit, den aktuellen Bearbeitungsstand der verschiedenen Prozesse im Lebenszyklus der Rollen einsehen zu können. Dies schließt eine Betrachtung früherer Rollenzuweisungen auf feingranularer Ebene ebenso mit ein, wie eine Analyse, wie sich zukünftige Änderungen an der Rollen- und Rechtestruktur auf das bestehende Rollenmodell auswirken würden. Die Komponente *security administration console* repräsentiert die grafische Bedienoberfläche für alle Komponenten dieses Arbeitsbereichs. Sie ist individuell an die Anforderungen der OIM-Benutzer anpassbar. Dazu führt OIM Ansichten (engl. *views*) ein, für die definiert werden kann,

welche Bereiche in ihnen dargestellt werden. Am Ende dieses Überblickskapitels wird in Information 20 die Liste der vordefinierten Ansicht dargestellt. Die letzten beiden Komponenten *service level agreements* und *key performance indicators* dienen zur Messung der *workflows*. Hierbei können klare Zeitgrenzen für die Durchführung einzelner Prozessschritte, aber auch Maßnahmen definiert werden, die im Fall von Überschreitungen der Indikatorwerte ergriffen werden sollen.

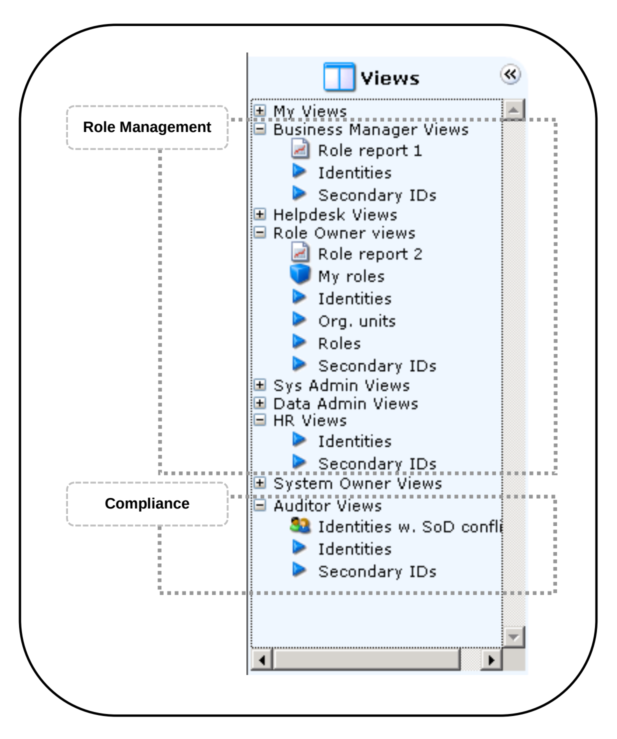
In der vorliegenden Version des Omada Identity Manager wird ein Authentifizierungsmodul für die Produkte SiteMinder der Firma CA und Tivoli Access Manager der Firma IBM angeboten. Beides sind kommerzielle Zugriffskontrollsysteme die durch das Modul *authentication* des zur integrierten Authentifizierung verwendet werden können. Zusätzlich dazu kann die Bedienoberfläche des Omada Identity Manager durch die Komponente *multi-language* sehr leicht lokalisiert und somit an die Sprachen der Benutzer des Systems angepasst werden. *Authentication* und *multi-lanuguage* sind der Vollständigkeit halber in Information 18 aufgeführt, werden im Fortgang dieses Kapitels aber nicht weiter betrachtet, weil sie außerhalb des Kerns dieser Arbeit liegen. Dies beendet die Übersicht über die Aufgabenbereiche und Komponenten des OIM. Um das Überblickskapitel zu vervollständigen, wird nun zunächst auf die Datentypen im Kontext der Rollen sowie die Beziehungen untereinander eingegangen und abschließend ein Einblick in die grafische Bedienoberfläche des Omada Identity Manager gegeben.



Information 19: State of the Art – Omada Identity Manager Data Type Relationship

Information 19 stellt die Datentypen des OIM im Hinblick auf die rollenbasierte Zugriffskontrolle dar. An dieser Stelle sei erwähnt, dass an dieser Stelle lediglich diejenigen Datentypen aufgeführt wurden, die im Zusammenhang mit Rollen stehen. Der zentrale Datentyp Rolle (engl. *role*) wird im Omada Identity Manager in unterschiedlichen Kontexten verwendet und hat demnach unterschiedliche semantische Bedeutungen. Dieser Bedeutungsunterschied drückt sich in Form des Rollentyps (engl. *role type*) aus. So kann eine Rolle sowohl als Einheit für geschäftliche Funktionen dienen, was der in dieser Arbeit verwendeten Geschäftsrolle entspricht, als auch technische Details kapseln. Dies steht in Bezug zu dem hier definierten Begriff Systemrolle. Darüber hinaus kann in OIM ein einzelnes Benutzerobjekt ebenfalls eine Rolle darstellen, was im Folgenden aber nicht weiter beleuchtet oder differenziert wird. In OIM existiert eine Vielzahl unterschiedlicher Hierarchien. So gibt es zunächst eine Hierarchie auf Ebene der Geschäftsstruktur, womit sich Niederlassungen, Zweigstellen oder Abteilungen abbilden lassen. Davon losgelöst existiert eine Rollenhierarchie bei Systemrollen. OIM unterstützt dabei nicht

nur den Aufbau einer Hierarchie, sondern auch die Vererbung von Rechten. Dies wird dadurch realisiert, dass Mitgliedschaften in Systemrollen an hierarchisch untergeordnete Systemrollen weitergegeben werden. Auf einer dritten, davon unabhängigen Abstraktionsebene wird jede Rolle eindeutig in einem Rollenordner (engl. *role folder*) abgelegt, der seinerseits ebenfalls in eine Hierarchie eingebunden ist. OIM bietet Einschränkungen (engl. *constraints*) für Rollen an, wodurch etwa das SoD-Prinzip umgesetzt wird. Ermöglicht wird diese Form der Einschränkung dadurch, dass explizit definiert werden kann, welche Rollen sich wechselseitig ausschließen. Diese *constraints* werden systemweit einmalig definiert, womit erreicht wird, dass sie sich auf das ganze Unternehmen auswirken und nicht umgangen werden können.



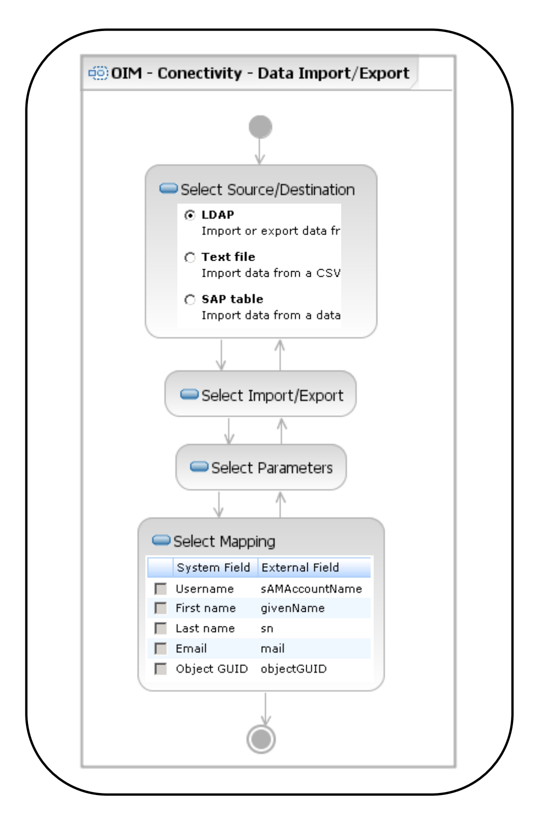
Information 20: State of the Art – Omada Identity Manager GUI

Abschließend soll ein kurzer Einblick in die grafische Bedienoberfläche gegeben werden. In Information 20 wird das Konzept der *view* dargestellt, welches im Omada Identity Manager verwendet wird, um die Bedienoberfläche für die unterschiedlichen Mitarbeiter anzupassen und ihnen den für ihre Rollen benötigten Ausschnitt des Gesamtsystems darzustellen. Diese Ansichten orientieren sich an den Aufgaben der Mitarbeiter im Umgang mit dem Rollenmanagementwerkzeug. Im oberen Bereich sind einige Sichten aufgezeigt, die im Zusammenhang zum ersten Arbeitsbereich *role management* stehen. Der untere Bereich stellt den Bezug zum zweiten Arbeitsbereich *compliance* her.

3.2.2 Der Arbeitsbereich *connectivity*

Nachdem der Arbeitsbereich *connectivity* im Überblickskapitel 3.2.1 bereits verbal erklärt wurde, soll nun eine prozessorientierte Einführung gegeben werden. Damit der Omada Identity Manager seine Funktionen im Rollenmanagement erbringen kann, müssen ihm die dazu benötigten Daten initial zugeführt werden und über die Lebenszeit der Rollen aktuell gehalten werden. Dies umfasst die Aufgaben des Arbeitsbereichs *connectivity*. In diesem Teilkapitel werden die beiden Komponenten des OIM, die diese Aufgaben erfüllen, einer genauen Betrachtung unterzogen. Der Omada Identity Manager verfügt über zwei unterschiedliche Mechanismen zur Interaktion mit den Unternehmenssystemen. Dies sind zum Einen ein Datenaustausch in Form eines Imports in OIM und eines anschließenden Exports, der die verarbeiteten Daten in die Unternehmenssysteme zurückführt und zum Anderen die Kommunikation über ein Provisionierungswerkzeug, was über Managementagenten realisiert wird. Zunächst wird der Prozess zum

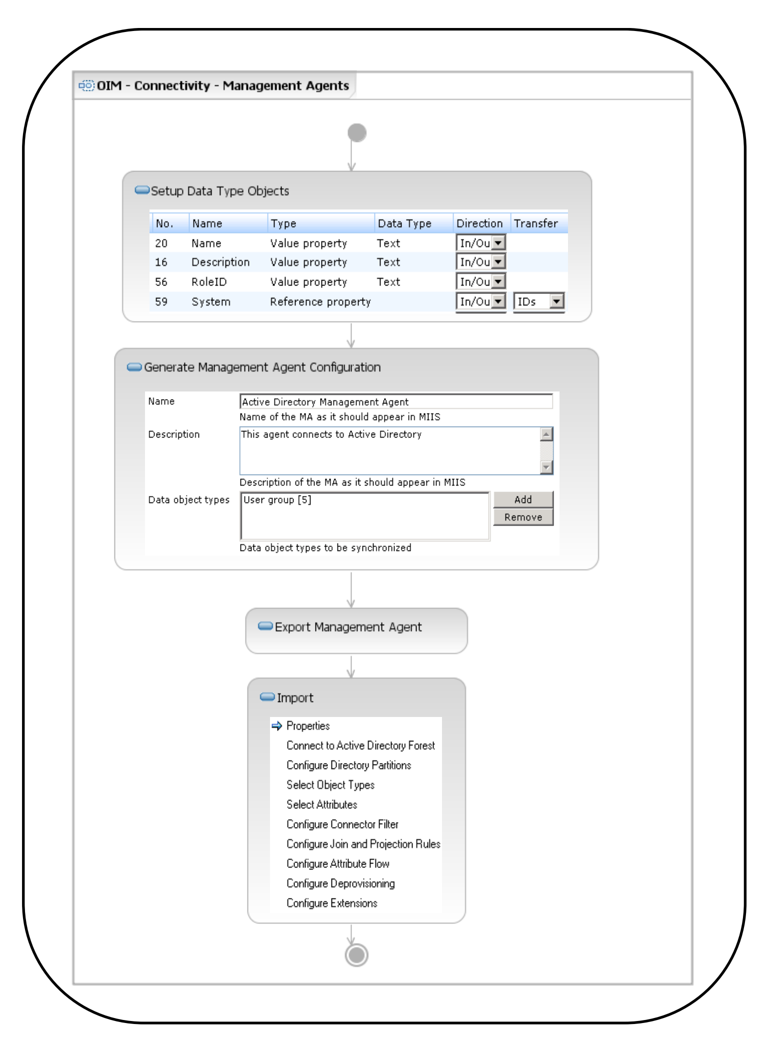
Import/Export verbildlicht, dargestellt in Information 21 und anschließend die Kommunikation in Form von Managementagenten in Information 22.



Information 21: State of the Art – Omada Identity Manager Data Exchange

Beim Anlegen von zu importierenden oder exportierenden Daten beginnt man damit, die Quelle und das Ziel der Daten zu spezifizieren. Der Omada Identity Manager bietet hierfür drei mögliche Verbindungspartner an: Eine direkte Kommunikation mit einem LDAP-konformen Verzeichnisdienst, einer SAP-Tabelle und einer manuell zu pflegenden CSV-Datei. Anschließend wird in der Aktivität *select import/export* angegeben, ob es sich bei dieser Datenkommunikation um einen Import in OIM oder einen Export aus OIM handelt. Man kann im folgenden Schritt zusätzliche Parameter angeben, wie etwa einen Zeitplan, wann die Aktion auszuführen ist und ob sie regelmäßig wiederholt werden soll. Auch kann hier spezifiziert werden, ob eine vollständige Übertragung aller Daten im Sinne eines kompletten Datenaustauschs, oder nur bisher nicht vorhandener Daten gewünscht ist. Des Weiteren muss angegeben werden, wie damit umgegangen werden soll, wenn Inkonsistenzen zwischen beiden Datenbeständen erkannt werden. Abschließend müssen die zu übertragenen Attribute ausgewählt und in Relation zu den entsprechenden Attributen im Zielsystem gesetzt werden.

Die zweite Art der Kommunikation verwendet den Microsoft Identity Lifecycle Manager (ILM) als Provisionierungsplattform. Hierfür sind Managementagenten nötig, die bei Aktualisierungen des Metaverzeichnisses des ILM aktiviert werden und dem Omada Identity Manager die geänderten Daten über eine Webservice-Schnittstelle zuführen. OIM unterstützt die grafische Erstellung von Managementagenten, was in Information 22 dargestellt wird.



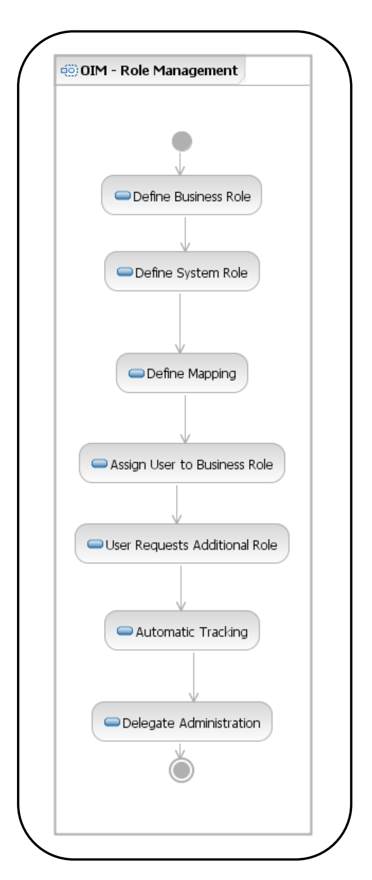
Information 22: State of the Art – Omada Identity Manager Management Agents

Der Prozess zur Verwendung von Managementagenten gestaltet sich etwas anders als beim Import/Export von Daten: Die Datenbasis für Agenten stellen die Datenobjekte von OIM dar. Diese bestehen aus einer Menge von Attributen, wobei für jedes Attribut separat spezifiziert werden kann, in welche Richtung es synchronisiert werden soll. Das Ergebnis dieses Ansatzes ist, dass die Datenobjekte mit einer Teilmenge ihrer Attribute komplett im Metaverzeichnis des ILM abgelegt werden. Für jedes zu synchronisierende Datenobjekt samt der soeben spezifizierten Attribute liefert OIM daraufhin einen eigenen Managementagenten zurück, der in den ILM importiert werden muss, damit die Synchronisation in beide Richtungen gewährleistet ist. Die Konsequenz dieses Ansatzes ist, dass beim Import des Agenten im Identity Lifecycle Manager ein Konfigurationsaufwand entsteht, da für jedes der Datenobjekte und dessen Attribute eine eigene Relation zu den Attributen der unterschiedlichen Unternehmenssysteme hergestellt werden muss.

3.2.3 Der Arbeitsbereich *role management*

Nachdem die Kommunikationsbeziehungen zwischen OIM und den Endsystemen beschrieben wurde, werden in diesem Teilkapitel die Komponenten des Arbeitsbereichs *role management* genau betrachtet. Um dies zu veranschaulichen, werden sie in einem Prozess dargestellt. Dazu wird zunächst die Relation zwischen Geschäfts- und Systemrollen im Omada Identity Manager betrachtet und anschließend ein Arbeitsablauf (engl. *workflow*) aufgezeigt. In diesem *workflow* wird initial eine Geschäftsrolle und eine Systemrolle definiert. Anschließend werden diese beiden Rollen in Relation zueinander gesetzt und ein Benutzer in die Rollenstruktur eingepflegt. Anhand dieses Benutzers werden die Verwaltungsaufgaben, die OIM im Zusammenhang mit

Rollen anbietet, vorgestellt. Dieser Prozess stellt den Bezug zwischen den Komponenten *role maintenance*, *role mapping*, *self service workflow*, *automation* und *delegated administration* aus Information 18 her, die zu Beginn dieses Kapitels vorgestellt wurden. Der *workflow* ist in folgender Abbildung dargestellt.

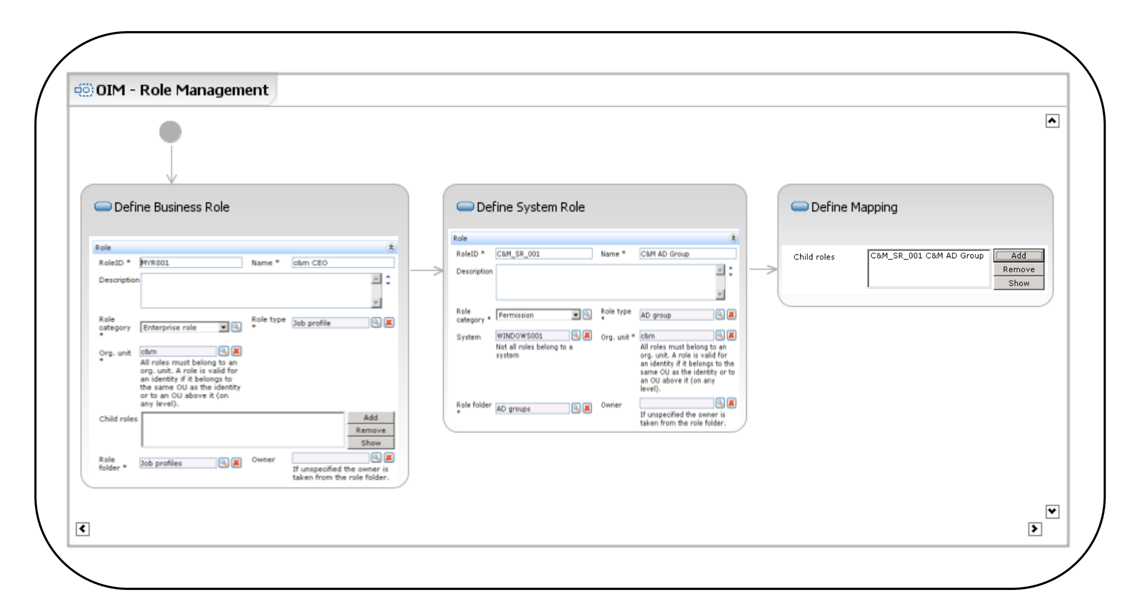


Information 23: State of the Art – Omada Identity Manager Role Management

Im Vergleich zum zugrundeliegenden RBAC-Standard aus [FS+01] interpretiert der Omada Identity Manager den Rollenbegriff deutlich freier und so kann eine Rolle neben einer Kapselung geschäftlicher oder technischer Rechte auch für einen Benutzer selbst stehen. Der semantische Unterschied zwischen diesen drei Rollenbegriffen wird durch ein entsprechendes Attribut innerhalb der Rolle festgelegt. Die Rollen, die im Omada Identity Manager verwendet werden, sind in mehrere unterschiedliche Rollentypen (engl. *role types*) unterteilt, deren Eigenschaften und Schemata sich jeweils von einer Vorlage ableiten, wie aus Information 19 ersichtlich ist. Durch die Vorlagen wird ermöglicht, gewisse Attribute oder Eigenschaften wie etwa die Vererbung von Rollen für die einzelnen Rollentypen zu definieren oder explizit zu unterbinden. Eine Rolle kann somit neben anderen Attributen insbesondere über eine Menge weiterer Rollen verfügen, wodurch die Hierarchiebildung auf Rollenebene technisch realisiert wird. Aufgrund der Mehrdeutigkeit des Rollenbegriffs ist in OIM eine Hierarchie sowohl auf geschäftlicher, als auch auf technischer Ebene möglich. Die Verknüpfung zwischen diesen beiden Ebenen wird dadurch hergestellt, dass eine Geschäftsrolle mit einer Menge von Systemrollen in Bezug gesetzt wird. Durch die Synchronisation mit dem Metaverzeichnis des ILM können bereits bestehende Rollen aus Endsystemen importiert und verwendet werden, falls diese bereits über Rollen verfügen. Die Festlegung der technischen Berechtigungen, die eine Systemrolle verkörpert, geschieht allerdings außerhalb von OIM, so dass Systemrollen lediglich als abstraktes Objekt vorgehalten werden. Hierarchien sind in OIM ein sehr weitgefasster Begriff: So gibt es zunächst eine Hierarchie auf Ebene der Organisation. Dadurch wird eine Unterteilung der gesamten Organisation in Abteilungen oder Zweigstellen ermöglicht. Davon unabhängig existiert, wie bereits angedeutet wurde, eine Hierarchie auf Rollenebene, was aufgrund des Rollenmodells in

OIM sowohl eine Hierarchie von Geschäftsrollen, Systemrollen und sogar eine Mischung von beidem ermöglicht. In OIM wird jede Rolle in Rollenordner eingeteilt, wie aus Information 19 hervorgeht. Dadurch ist eine weitere Strukturierung unabhängig von der Organisation oder Rollenhierarchien möglich. Diese dient dazu, die unterschiedlichen Ausprägungen von Rollen zu unterteilen, etwa in Geschäftsrollen und Systemrollen, die ihrerseits nochmals nach der Zugehörigkeit zu unterschiedlichen Endsystemen angeordnet sind. Die Verknüpfung von Geschäfts- und Systemrollen ist in OIM unmittelbar möglich, da jede Rolle mit beliebig vielen weiteren Rollen direkt verknüpft sein kann. In OIM verfügen die Datenobjekte Rolle und Organisationseinheit über einen Besitzer, womit die Administration dieser Objekte delegiert und die Verantwortlichkeit für diese Objekte dezentralisiert werden kann.

OIM passt sich den Bedürfnissen der Benutzer durch unterschiedliche Ansichten (engl. *views*) an, was im Überblickskapitel bereits angesprochen wurde. Diese Sichten richten sich an die unterschiedlichen Benutzergruppen, die bei der Verwaltung von Rollen unterschiedliche Arbeiten ausführen. Der Arbeitsprozess zum Anlegen einer Rolle erstreckt sich somit über mehrere Befugnisgrenzen, was in OIM dadurch zum Ausdruck kommt, dass die einzelnen Arbeitsschritte in unterschiedlichen Sichten aufgeführt werden. Zunächst beginnt der Rollenverantwortliche, die Geschäfts- und Systemrollen in OIM anzulegen. Dabei werden sie bereits einer Abteilung oder allgemein einer Geschäftseinheit zugeordnet. Auch werden hier die Rollen in einen Rollenordner eingeordnet. Anschließend muss eine Verknüpfung zwischen diesen beiden Rollen hergestellt werden. Dies geschieht dadurch, dass die Systemrolle in die Liste der mit der Geschäftsrolle verknüpften Rolle aufgenommen wird. Dieser Teil des Arbeitsablaufs ist in Information 24 dargestellt.



Information 24: State of the Art – Omada Identity Manager Role Management (1)

Nachdem die Rollenstruktur aufgebaut wurde, folgt die Einteilung der Benutzer in Geschäftsrollen. Dies geschieht im Allgemeinen nicht mehr in der Verantwortung des Rollenverantwortlichen, sondern ist eher die Aufgabe eines Abteilungsleiters. Es wird davon ausgegangen, dass die entsprechenden Benutzerkonten zu diesem Zeitpunkt bereits im Verzeichnisdienst angelegt und OIM über die Komponente *connectivity* zugeführt wurden. Ferner ist die Abteilungszugehörigkeit des Benutzers beispielsweise durch die Personalabteilung bereits vorgegeben. Der für die Abteilung Verantwortliche bestimmt im Ansatz, den OIM verfolgt, die Einteilung des Benutzers in Rollen und eine Organisationseinheit. In der *view* des Abteilungsleiters wird der neue Benutzer daraufhin aufgeführt. Dies geschieht durch einen automatischen Arbeitsablauf (engl. *workflow*), der in regelmäßigen Abständen nach neuen oder geänderten Daten in den Unternehmenssystemen sucht. Der Rollenverantwortliche wird von OIM daraufhin über einen neuen Mitarbeiter in seiner Abteilung informiert, so dass er diesen in eine Geschäftsrolle einteilen kann.

Im Zuge dieser Einteilung werden vom Omada Identity Manager alle Systemrollen sowie weitere Geschäftsrollen ermittelt, die sie durch die Rollenhierarchie erbt. Schließlich werden diejenigen Rollen ermittelt, die mit der Organisationseinheit verknüpft sind, in der der Benutzer eingeteilt wurde. Der neue Mitarbeiter wird dann durch OIM automatisch in alle diese Rollen eingetragen. An dieser Stelle sei erwähnt, dass das Prinzip *separation of duties* hierbei auch beachtet wird. Erhielte der Mitarbeiter, der durch den Abteilungsleiter einer Organisationseinheit sowie einer Geschäftsrolle zugeordnet wird, mehrere Rollen, die sich wechselseitig ausschließen, werden diese Rollen konsequenterweise nicht zugewiesen. Da diese Komponente zum Arbeitsbereich *compliance module* gehört, wird sie im folgenden Kapitel näher beleuchtet. Eine Komponente des OIM ermöglicht es jedem einzelnen Benutzer selbständig, Änderungen an der Rolleneinteilung zu initiieren, was im Wirkbetrieb sehr viel Zeit spart. Dies betrifft die dauerhafte Änderung aktuell zugewiesener Rollen sowie eine zeitlich befristete Weitergabe von Rollen an eine andere Person. Die Komponente, die dabei Änderungswünsche mitverfolgt und an die entsprechenden Bearbeiter weiterleitet, ist das *automatic tracking*. Sie identifiziert das Personal, welches für die Änderungen autorisiert ist und informiert sie über die Änderungswünsche. Somit wird sichergestellt, dass die dezentral initiierten Änderungen an den dafür verantwortlichen Mitarbeiter delegiert und von ihm angenommen oder abgewiesen werden können. Dieser zweite Teil des dargestellten Arbeitsablaufs ist in Information 25 dargestellt und beschließt den Arbeitsbereich *role management*.



Information 25: State of the Art – Omada Identity Manager Role Management (2)

3.2.4 Der Arbeitsbereich *compliance*

Nachdem im vorangegangenen Kapitel derjenige Bereich des Omada Identity Manager betrachtet wurde, der sich mit dem Rollenmodell und dessen Wartung im Wirkbetrieb befasst, folgt nun ein detaillierter Einblick in den Arbeitsbereich *compliance*. Dieser zweite Arbeitsbereich von OIM befasst sich insgesamt betrachtet mit der Einhaltung von Policys sowie der Messung der *workflows*. Damit können Schwachstellen im Prozess erkannt, oder zeitliche Verzögerungen identifiziert werden. Wie im Überblickskapitel 3.2.1 erwähnt wurde, bietet OIM für die Bereitstellung dieser Funktionen mehrere Komponenten an, die im Folgenden einzeln betrachtet werden.

Die Komponente *attestation* befasst sich mit den Berechtigungen, die eine Rolle im Laufe ihrer Existenz besitzen kann. Dabei werden mehrere Möglichkeiten angeboten, die Zuweisung von Berechtigungen zu überprüfen. Wie schon im ersten Arbeitsbereich *role management* erleichtern auch hier die *workflows* diese Arbeiten. So ist es etwa möglich, dass ein Rollenverantwortlicher die Berechtigungen seiner Benutzer in regelmäßigen Abständen überprüft, verifiziert und

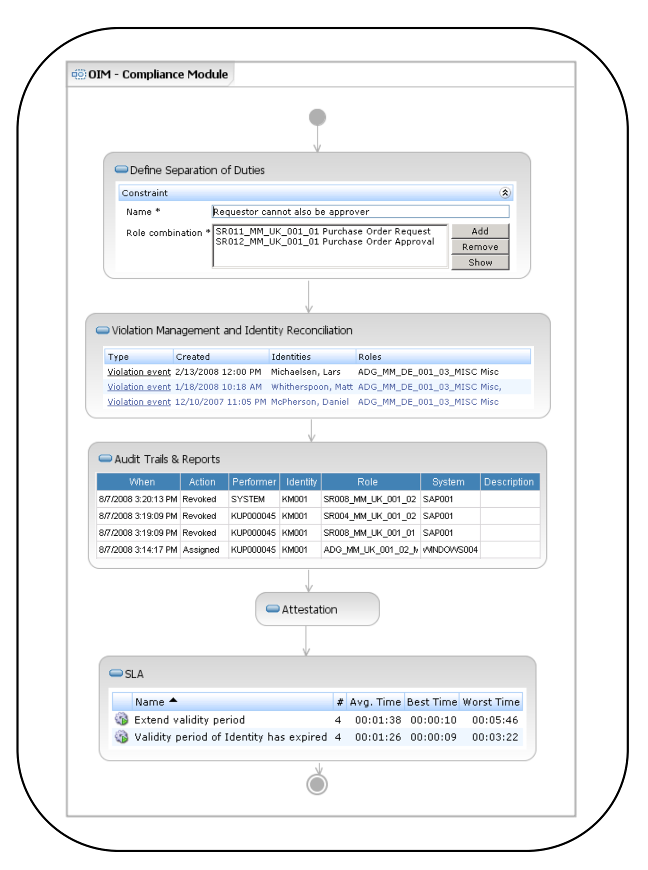
gegebenenfalls anpasst. An dieser Stelle wird das zentrale Konzept der individuellen *views* deutlich: Je nach Befugnis des Verantwortlichen wird individuell festgelegt, welche Eigenschaften von ihm eingesehen und dadurch auch aktiv analysiert werden können. Der Besitzer einer Rolle kann beispielsweise sämtliche Veränderungen dieser Rolle für das gesamte Unternehmen einsehen und vornehmen, ein Manager hingegen nur diejenigen Rolleneigenschaften, die in seinem Verantwortungsbereich liegen. Im Gegensatz zum Rollenbesitzer kann der Manager diese Eigenschaften aber von allen Rollen beeinflussen, die die Angestellten seiner Abteilung besitzen und nicht nur von einer Rolle. Eine wesentliche Voraussetzung bei der Einhaltung von Policys ist, dass die zugrunde liegende Datenstruktur konsistent ist. Wird auf einem inkonsistenten Datenbestand operiert, macht die Frage nach der Einhaltung von Policy-Vorgaben kein Sinn, weil die Vorgaben aufgrund von Inkonsistenzen nicht gewährleistet werden können. Aus diesem Grund muss sichergestellt werden können, dass das Rollenmanagementwerkzeug ein konsistentes Gesamtbild des Unternehmens besitzt und beibehält. Werden nun Änderungen in einem der Endsystemen direkt vorgenommen, anstatt im Omada Identity Manager, werden diese Änderungen erkannt und entsprechende Mechanismen eingeleitet, wie etwa das Rückgängigmachen dieser Änderungen und das Informieren eines Systemverantwortlichen über die unbefugte Änderung. Hierbei müssen dem Omada Identity Manager allerdings die zu überwachenden Attribute bekannt sein, was in den Aufgabenbereich *connectivity* fällt, da das Werkzeug die Änderungen sonst nicht erkennen und entsprechende Maßnahmen ergreifen kann. Eine weitere Komponente in OIM befasst sich mit Änderungen von Zugriffsrechten bei Benutzern und überwacht dabei speziell die Eigenschaften der Benutzer im Verzeichnisdienst. Sie wird aus diesem Grund als *identity reconciliation* bezeichnet. Durch diese Komponente wird die Konsistenz der Benutzerprofile gewährleistet, die sowohl im Unternehmensverzeichnisdienst, als auch im Metaverzeichnis des OIM vorgehalten werden.

Das Prinzip *separation of duties* (SoD) verkörpert Regeln, wonach es Rechte gibt, die sich gegenseitig ausschließen und die demnach nicht an ein und dieselbe Rolle vergeben werden können. In OIM wird dieses Prinzip in Form von Einschränkungen (engl. *constraints*) realisiert, die über eine Menge an Rollen verfügen, die sich gegenseitig ausschließen, wie in Information 19 dargestellt ist. Sobald in OIM eine Änderung an den Rollenzuweisungen erkannt wird, werden die bestehenden *constraints* in einem Hintergrundprozess gesammelt und überprüft, ob sich durch die geänderten Rollenzuweisungen Policy-Verletzungen ergeben. Die Änderung ist nur dann erfolgreich, wenn keine Policy-Verletzungen vorhanden sind. Dabei wirken sich Einschränkungen auch auf die bisher erteilten Recht aus: Denjenigen Benutzern, die von einer neu angelegten Einschränkung betroffen sind, werden die Rollen entzogen und vermerkt, dass dies aufgrund einer Einschränkung der Fall ist. Fordert ein Benutzer eigenständig eine Rolle an, wird er beim Auftreten eines *constraint* darauf hingewiesen und die Rolleneinteilung wird nicht durchgeführt. Ein Bericht über aktuelle Verletzungen dient dazu, die Verletzungen für die technische Administration sichtbar zu machen. Auch dient dies zur Planung einer Umstellung der Rechtestruktur: Es kann einerseits beobachtet werden, welche Rechteverletzungen gehäuft auftreten, was ein Indiz dafür sein kann, dass der Rechteumfang von gewissen Rollen nicht angemessen ist und ausgeweitet werden muss. Auch kann durch den Bericht analysiert werden, wie sich geplante Änderungen im Wirkbetrieb auswirken würden und welche Verletzungen diese Änderungen nach sich ziehen.

Da der Omada Identity Manager alle Ereignisse intern abspeichert, bietet er für jeden Benutzer die Möglichkeit, sich alle mit ihm verknüpften Ereignisse anzeigen zu lassen. Somit können die Benutzer auf dem aktuellen Stand gehalten werden, was die Prozesse im Rollenmanagement angeht. Diese Komponente kann sowohl den aktuellen Bearbeitungsstand anzeigen, als auch Berichte aus der Vergangenheit darzustellen. Dies ermöglicht, Verletzungen oder geänderte Zugriffsmuster von Benutzern auch in der Vergangenheit aufzuspüren um die Rollenstruktur an die geänderten Gegebenheiten anzupassen. Um sich hier auch auf spezielle Ereignisse, Zeiträume oder andere Eigenschaften zu beschränken, lässt sich der Datenbestand vor der Berichterstellung filtern.

Die Bedienoberfläche dieses Arbeitsbereichs, als *security administration console* bezeichnet, kombiniert die Statistiken zur Einhaltung von Policys mit Metriken zur Messung der Prozesse, auf die zum Schluss eingegangen wird. Sie stellt die zentrale Bedienoberfläche des OIM im Bezug auf *compliance* dar.

Neben den Statistiken zur Einhaltung von Policys bietet der Omada Identity Manager die Möglichkeit, Zeitgrenzen bei der Durchführung von Prozessen zu definieren sowie eine Überprüfung des aktuellen Stands von Prozessen, die sich gerade in der Ausführung befinden. Dadurch ist es für das verantwortliche Personal möglich, die Leistungsindikatoren von Prozessen auf feingranularer Ebene zu überwachen, Schwachstellen oder Engpässe zu entdecken und Garantien für diese Prozesse zu gewähren. Im verbleibenden Teilkapitel werden diese Komponenten anhand eines Prozesses erklärt, der sich an den Prozess aus dem letzten Teilkapitel anschließt und die Komponenten prozessbezogen darstellt. Er ist dargestellt in Information 26.

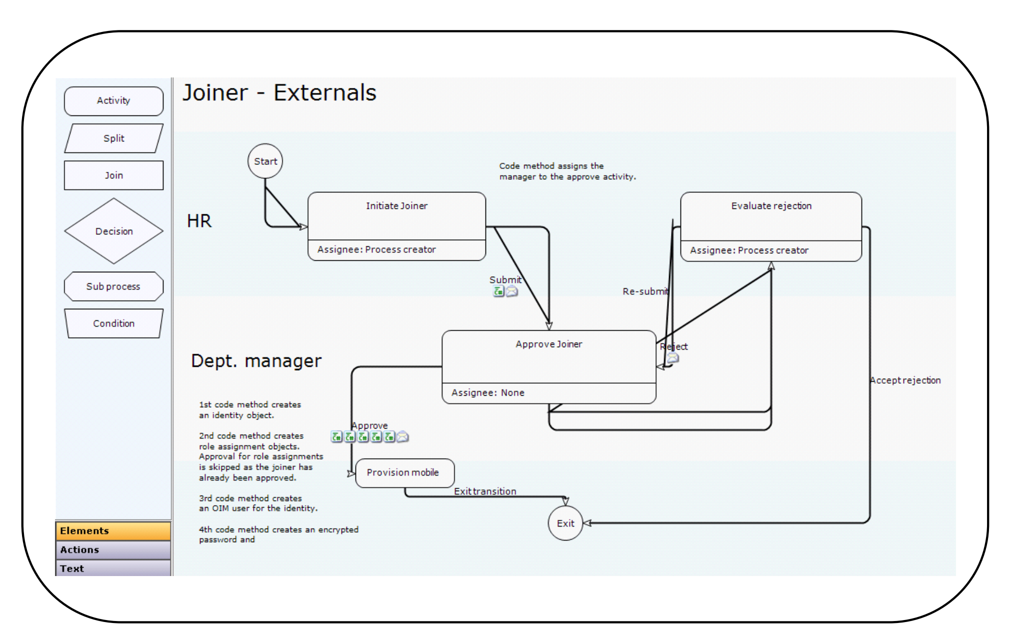


Information 26: State of the Art – Omada Identity Manager Compliance Module

In diesem Prozess wird zunächst eine *separation of duty*-Beschränkung für zwei Rollen definiert. Anschließend wird überprüft, ob von ihr aktuell Benutzer betroffen sind und gegebenenfalls die Rollen aufgeführt, die die SoD-Verletzung verursachen. Den identifizierten Benutzern wird daraufhin sofort die Mitgliedschaft in den konfliktbehafteten Rollen entzogen und der Rollenverantwortliche darüber in Kenntnis gesetzt. Im nächsten Schritt ist in Information 26 die Ansicht eines Benutzers dargestellt, der die Historie seiner Rollenzuteilungen einsieht. Dies erhöht die Transparenz der Gesamtumgebung. Daraufhin können die Rechte des Benutzers analysiert und gegebenenfalls angepasst werden. Sollte dieser Schritt außerhalb von OIM vorgenommen werden, wird dies einem Verantwortlichen signalisiert. Im letzten Prozessschritt sind die Prozessindikatoren für zwei *workflows* dargestellt, aus denen die Leistung der *workflows* abgelesen werden kann. Diese Kennzahlen werden zur Messung der Dienstgüte und der daraus abgeleiteten Qualitätsgarantie (engl. *service level agreement*) herangezogen. Im Beispiel sind dies etwa die Anzahl der Ausführungen oder die durchschnittliche Bearbeitungszeit.

3.2.5 Automatische Abläufe und werkzeugunterstützte Bewilligungen

OIM bietet eine eigene grafische Bedienoberfläche zur Erstellung von Arbeitsabläufen (engl. *workflows*). Diese Arbeitsabläufe wurden schon vereinzelt erwähnt, denn sie sind ein elementarer Bestandteil von OIM. Durch sie werden alle Automatismen in OIM realisiert. Aus diesem Grund soll der Designer zum Ende des Einführungskapitels zum Omada Identity Manager exemplarisch präsentiert werden. Hierbei bietet OIM eine eigene grafische Modellierungssprache an um die *workflows* zu spezifizieren. In Information 27 ist ein Beispielprozess abgebildet, der das Hinzufügen eines neuen Mitarbeiters in OIM teilautomatisiert. Angesteuert wird dieser Prozess, sobald im Verzeichnisdienst ein neuer Mitarbeiter erkannt wird. Anschließend wird eine neue Identität erzeugt, die die Eigenschaften zugewiesen bekommt, die das Mitarbeiterobjekt im Verzeichnisdienst aufweist. Dies sind neben dem Namen auch Eigenschaften wie etwa die Abteilungszugehörigkeit. Anschließend wird der Prozess an den Abteilungsleiter derjenigen Abteilung delegiert, in der der neue Mitarbeiter arbeiten wird. Er muss dem neuen Mitarbeiter eine Rolle im Sinne des OIM zuweisen, oder ihn abweisen. Im ersten Fall wird der Prozess an die IT-Abteilung weitergereicht, die dem Mitarbeiter ein Geschäftsmobiltelefon ausstellt. Schließlich wird ein automatisches Passwort erzeugt, welches dem Abteilungsleiter zugestellt wird. Im zweiten Fall wird der Prozess mit einer Begründung für die Abweisung an die Personalabteilung zurückgeleitet.



Information 27: State of the Art – Omada Identity Manager Workflow Designer

Dies beendet den Einblick in das Rollenmanagementwerkzeug Omada Identity Manager. Im folgenden Kapitel wird mit dem Role Manager der Firma Sun ein zweites Werkzeug präsentiert, welches im Gegensatz zum Omada Identity Manager schon länger auf dem Markt präsent ist und über eine Komponente zur Entwicklung von Rollen verfügt. Der *role engineering*-Ansatz, der im Sun Role Manager verfolgt wird, ist das *role mining*, welches in den Kapiteln 2.3 als typische Aufgabe von Rollenmanagementwerkzeugen und in Kapitel 3.1.1 als formales Modell bereits vorgestellt wurde.

3.3 Rollenbasierte Zugriffskontrolle im Sun Role Manager

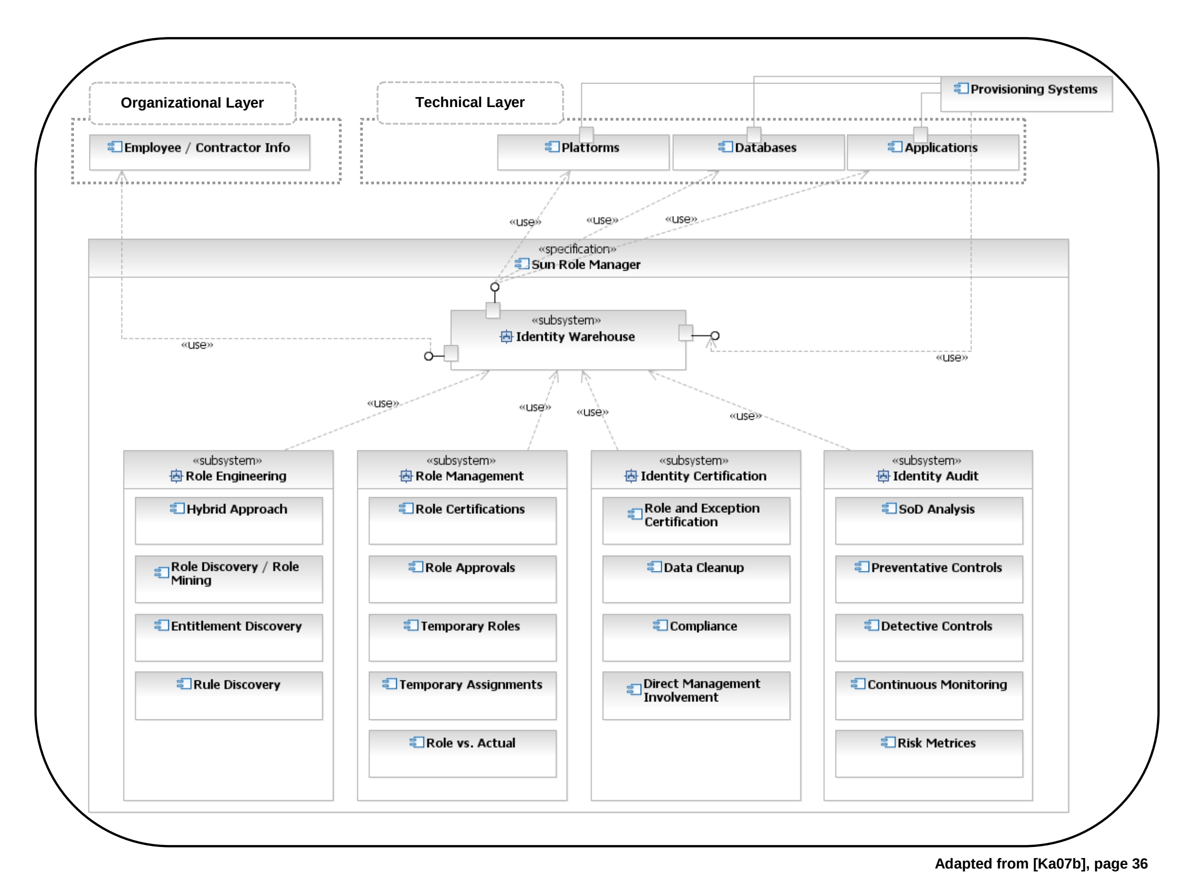
Das Produkt Sun Role Manager (SRM) basiert auf dem Rollenmanagementwerkzeug RBACx der Firma Vaau Incorporated. Sie wurde am 15. Februar 2008 von der Firma Sun Microsystems aufgekauft und RBACx wurde in Role Manager umbenannt, um es in die Produktlinie „Identity Management“ von Sun zu integrieren [Sun08a]. Zu diesem Zeitpunkt gehörte die Firma Vaau

mit seinen 60 Angestellten bereits zu den führenden Anbietern von Rollenmanagementwerkzeugen. Dies kann dadurch belegt werden, dass Vaau über eine langjährige Erfahrung mit Kunden aus unterschiedlichen Branchen verfügt, die RBACx produktiv einsetzen [Ka07b].

In dieser Arbeit wird die Version 4.0 von SRM eingesetzt. Der Funktionsumfang von SRM kann in drei wesentliche Aufgabenbereiche aufgeteilt werden, die alle auf einer zentralen Datenbank innerhalb des Role Manager operieren. Im folgenden Teilkapitel wird zunächst die Architektur des SRM sowie die von ihm verwendeten Datentypen erklärt. Die Datenbank und die drei Arbeitsbereiche werden kurz erklärt, durch ein Schaubild illustriert und jeweils in Bezug gesetzt zu den Aufgabenbereichen von Rollenmanagementwerkzeugen aus Kapitel 2.3. In den folgenden Teilkapiteln wird ein genauerer Einblick in diese Arbeitsbereiche gegeben, der die darin enthaltenen Funktionen anhand eines Prozesses illustriert. Auf Basis der daraus gewonnen Erkenntnisse wird das Produkt in Kapitel 6.3 anhand eines dort entwickelten Kriterienkatalogs bewertet. Das Ziel dieser Aufteilung ist es, zunächst einen abstrakten Überblick über das Produkt Sun Role Manager zu geben und anschließend die einzelnen Aufgabenbereiche im Detail zu beleuchten. Dabei soll der Funktionsumfang von SRM wertungsfrei dargestellt werden.

3.3.1 Architektur, Aufgabenbereiche und Datentypen des Sun Role Manager

Die drei Aufgabenbereiche des Role Manager operieren auf einer zentralen, SRM-internen Datenbankkomponente, dem sogenannten *identity warehouse*. Bei den drei Bereichen handelt es sich um die Verwaltung von Rollen (engl. *role engineering*, *role management*), die Zuweisung von Rollen (engl. *identity certification*) und das Überwachen der Zuweisungen (engl. *identity audit*). Der Role Manager verfügt über einen Mechanismus, der dem *identity warehouse* regelmäßig Daten aus den angeschlossenen Endsystemen zuführt. Die Daten können dabei sowohl technische Informationen wie etwa Zugriffsrechte, Gruppenmitgliedschaften oder Ähnliches enthalten, als auch Geschäftsinformationen, wie etwa eine Geschäftsstruktur oder Geschäftshierarchien. Dadurch, dass die zu importierenden Daten selbständig spezifiziert und einzelne Attribute und Eigenschaften selektieren werden können, dient das *identity warehouse* als Metaverzeichnis für die Komponenten des SRM. Es stellt den zentralen Konsolidierungspunkt der Unternehmensdaten dar, die für RBAC relevant sind [Ka07b, Sun08a].



Adapted from [Ka07b], page 36

Information 28: State of the Art – Sun Role Manager Component Architecture

Information 28 schildert, welche Bereiche des Rollenmanagements von SRM abgedeckt werden. Dabei sind die Geschäfts- und die technische Ebene eingetragen, um den Bezug zu dem für diese Arbeit grundlegenden Schaubild in Information 1 herzustellen. Wie bereits angedeutet, ist die zentrale Komponente des SRM das *identity warehouse*, ein Metaverzeichnis, welches den Komponenten von SRM zugrunde liegt und die Schnittstelle zu den unterschiedlichen Unternehmenssystemen darstellt [Sun08a]. Diese Datenbank wird somit von den drei Arbeitsbereichen *role engineering/role management*, *identity certification* und *identity audit* gleichermaßen verwendet. In den folgenden Unterkapiteln werden die in ihnen enthaltenen Funktionen prozessorientiert veranschaulicht. SRM kann in zwei unterschiedlichen Varianten in einer Unternehmensstruktur eingesetzt werden. Zum Einen kann es als alleinstehende Variante (engl. *stand-alone*) betrieben werden, so dass die Daten der Endsysteme dem *identity warehouse* manuell zugeführt werden müssen und zum Anderen als integrierte Lösung mit einer Identitätsmanagementanwendung, in der beispielsweise eine Provisionierungsplattform die Kommunikation zu den Endsystemen realisiert [Ka07b]. Um den Gegensatz zum Omada Identity Manager zu unterstreichen, wird hier die alleinstehende der integrierten Variante vorgezogen. Wie man aus der Grafik erkennen kann, bezieht SRM die Daten aus unterschiedlichen Datenquellen, kann die daraus abgeleitete Rollenstruktur und Policys aber nur im Falle einer integrierten Lösung unter Zuhilfenahme einer Provisionierungsplattform automatisiert in die Endsysteme zurückführen. Im *stand-alone*-Betrieb müssen diese Daten aus SRM durch einen manuellen Export an diese Systeme zurückgeführt werden. In der integrierten Variante mit Identitätsmanagementsystem, wie beispielsweise dem Sun Identity Manager als Provisionierungsplattform, ist der Prozess des Rückführens von Rollen und Policys somit komplett automatisierbar.

Der erste Arbeitsbereich umfasst Prozesse, die sich mit dem Definieren und der anschließenden Verwaltung der Rollen im produktiven Einsatz beschäftigen (engl. *role engineering*, *role management*). Zum Definieren von Rollen bietet SRM *data mining*-Algorithmen an, die auf der Basis von Benutzerberechtigungen arbeiten. Neben diesem automatisierten Ansatz ist es auch mög-

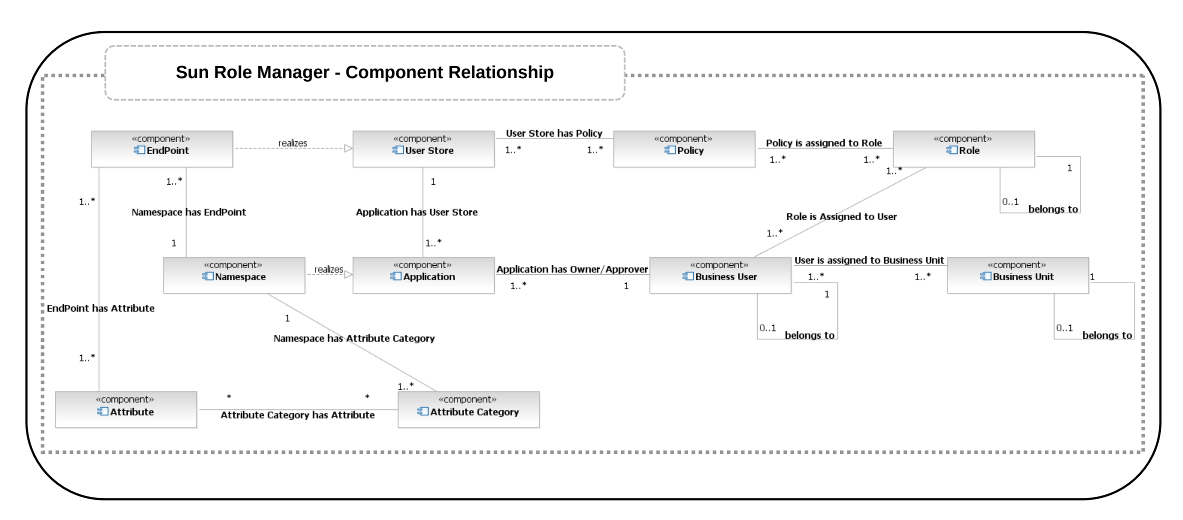
lich, die Definition manuell vorzunehmen. SRM verfolgt hierbei ein hybrides Vorgehen und bezieht damit sowohl Geschäfts- als auch Systemrollen in den Prozess mit ein. Die Verwaltung dieser Rollen geschieht über die einheitliche Web-Oberfläche des SRM [Ka07b]. Somit deckt dieser erste Bereich des Role Manager die Bereiche *business role modeling*, *business role management*, *IT role modeling*, *IT role management* und *role modeling and discovery* aus Information 11 ab.

Im zweiten Bereich befinden sich Prozesse, die sicherstellen, dass die zugewiesenen Berechtigungen und Mitgliedschaften eines Benutzers korrekt sind (engl. *attestation*). Der SRM ermöglicht es einem Rollenverantwortlichen wie etwa einem Abteilungsleiter oder Manager, diese Zuweisungen für seine Mitarbeiter auf feingranularer Ebene zu kontrollieren. Realisiert wird dies in zwei Schritten: Zunächst kann für jede Verbindung zu einer Datenquelle explizit angegeben werden, welche Attribute von dort in das *identity warehouse* importiert werden. Anschließend kann für jedes Attribut angegeben werden, ob es als zertifizierbares Attribut in SRM verwendet werden kann, wodurch ausgedrückt wird, dass es an Rollen zugewiesen werden kann. Die Möglichkeit der attributbasierten Zuweisung ist insbesondere beim Einsatz von Systemen wichtig, die ein sehr umfangreiches und feingranulares Autorisierungsmodell besitzen. Zusätzlich bietet das System die Möglichkeit zum Weiterleiten einer Zuweisungsaufgabe an eine andere Person, die nach Ablauf einer Frist automatisch an die ausstehende Arbeit erinnert wird. Dies stellt sicher, dass die Anfrage nicht vergessen wird. Da ein Unternehmen fortlaufend Änderungen personeller oder organisatorischer Art unterworfen ist, bietet SRM neben automatischen Erinnerungen für Rollenzuweisungen die Möglichkeit an, Berechtigungsänderungen auf einen bestimmten Zeitpunkt in der Zukunft zu verlegen, um den Arbeitsprozess erst zu diesem Zeitpunkt automatisch anzustoßen. Dadurch ist es zum Beispiel möglich, Änderungen in regelmäßigen Abständen zu planen oder etwa Überprüfungen an Berechtigungen vorzunehmen [Ka07b, Sun08a]. Da sich diese Aufgaben alle mit dem Überprüfen von Rollenzuweisungen von einer Menge an Mitarbeitern beschäftigt, werden sie im Unternehmenskontext eher von einer Person mit Personalverantwortung durchgeführt. Deswegen ist dieser zweite Arbeitsbereich des Role Manager auch eher der Geschäftsebene zuzuordnen. Die Funktion *identity certification* befindet sich in Information 11 in der Schnittmenge aus *attestation and compliance*, *policy management und role reconciliation*. Um die Sichtweise der Geschäftsebene zu unterstützen, bietet der Role Manager über die Benutzeroberfläche die Möglichkeit, die Zuweisungen namentlich zu verändern, was die Lesbarkeit im Wirkbetrieb wesentlich erhöht und somit den Verantwortlichen den Umgang mit diesem Werkzeug erleichtert. Weil die Benutzer wie hier beschrieben durch selbst gewählte Bezeichner von technischen Details abstrahieren können, wird die Bedienbarkeit des Werkzeugs erleichtert.

Der dritte Bereich des Sun Role Manager beschäftigt sich mit der Erkennung und Behebung von Ausnahmen und Policy-Verletzungen. SRM verwendet Policys (engl. *policies*) als Mechanismus zur Steuerung und Überwachung von Rollen. So wie bei den anderen beiden Bereichen stellt das *identity warehouse* auch hier sowohl die Datenquelle, aber auch den Speicherort der Ergebnisse dar. In diesem dritten Bereich sind das im Speziellen die im Wirkbetrieb aufgezeichneten Ausnahmen und Verletzungen. Dadurch bietet sich hier die Möglichkeit, diese Ereignisse in Echtzeit zu überwachen, aber ebenso im Nachhinein Sicherheitsrisiken oder Zugriffsmuster zu entdecken und dadurch die Umgebung dynamisch an die Änderungen anzupassen. Das Ziel der *identity audit*-Komponente ist das Sicherstellen, dass ein Benutzer nur über die Geschäfts- und Systemrollen verfügt, die für die Erledigung seiner Aufgaben benötigt werden. Dazu zählt auch, dass die Granularität der definierten Rechte, die durch das Rollenmodell vorgegeben ist, an die jeweils aktuellen Richtlinien im Unternehmen angepasst werden kann. Die *identity audit*-Komponente ermöglicht es dazu einerseits, einzelne Benutzer zu analysieren, um Verletzungen (engl. *violations*) etwa des SoD-Prinzips aufzudecken und andererseits bietet sie eine Verwaltungsseite an, um die Verletzungen und Ausnahmen (engl. *exception*) am Gesamtsystem in Echtzeit darzustellen. Hierzu werden die einzelne Verletzungen in unterschiedliche Kategorien oder Häufigkeitsklassen eingeteilt und aufsummiert angezeigt. Durch das spätere Speichern der Ereignisse in der Datenbank ist es möglich, einen Blick in die Vergangenheit

zu werfen, um Fehler im Nachhinein ausgiebig zu analysieren. Hier kommt SRM gesetzlichen Verpflichtungen nach, denen Unternehmen unterworfen sind (vgl. SOX, KonTraG). Im Gegensatz zur Analyse der Vergangenheit werden für die Echtzeitanalyse Überwachungsrichtlinien (engl. *audit policy*) benötigt, die aus Filterbedingungen (engl. *rule*) zusammengesetzt sind. Diese Überwachungsrichtlinien besitzen einen Verantwortlichen, dessen Aufgabe es ist, sich Verletzungen dieser speziellen Überwachungsrichtlinie anzunehmen. Ebenso kann hier eine Zeitspanne spezifiziert werden, nach deren Ablauf eine Erinnerungsnachricht verschickt wird, um sicherzustellen, dass die aufgetretene Verletzung behandelt wird. Am Ende des Lebenszyklus einer Überwachungsrichtlinie steht dann entweder das Weiterleiten an eine andere Person, die die Verletzung behandelt, oder das Abschließen der Verletzung, wenn die Ursache gefunden und behoben wurde [Ka07b, Sun08a]. Im Bezug auf die Aufgabenbereiche von Rollenmanagementwerkzeugen aus Information 11 spiegelt dieser Bereich durch das aktive Überwachen die Aufgabe *activity monitoring* wider. Da die Ergebnisse dieser Analysen unmittelbar in die Umgebung zurückfließen und sich dadurch das Rollenmodell anpasst, beinhaltet dieser Bereich zusätzlich dazu auch Aufgaben aus dem Bereich *role reconciliation*.

Zusammengefasst versteht der Sun Role Manager den Begriff Rollenmanagement als eine Kombination aus *discovery*, *design*, *engineering*, *optimization* und *lifecycle management* sowie den dazugehörigen Regeln in Form von Policys [Ka07b]. Er wendet *data mining*-Algorithmen auf sein Metaverzeichnis an, um dadurch Zugriffsrechte herauszuarbeiten und schlägt potentielle Geschäfts- und Systemrollen sowie konkrete Policys zur Steuerung und Kontrolle vor. Nachdem nun ein Überblick über die Architektur des Sun Role Manager gegeben wurde, illustriert Information 29 die verwendeten Datentypen sowie deren Beziehungen zueinander.

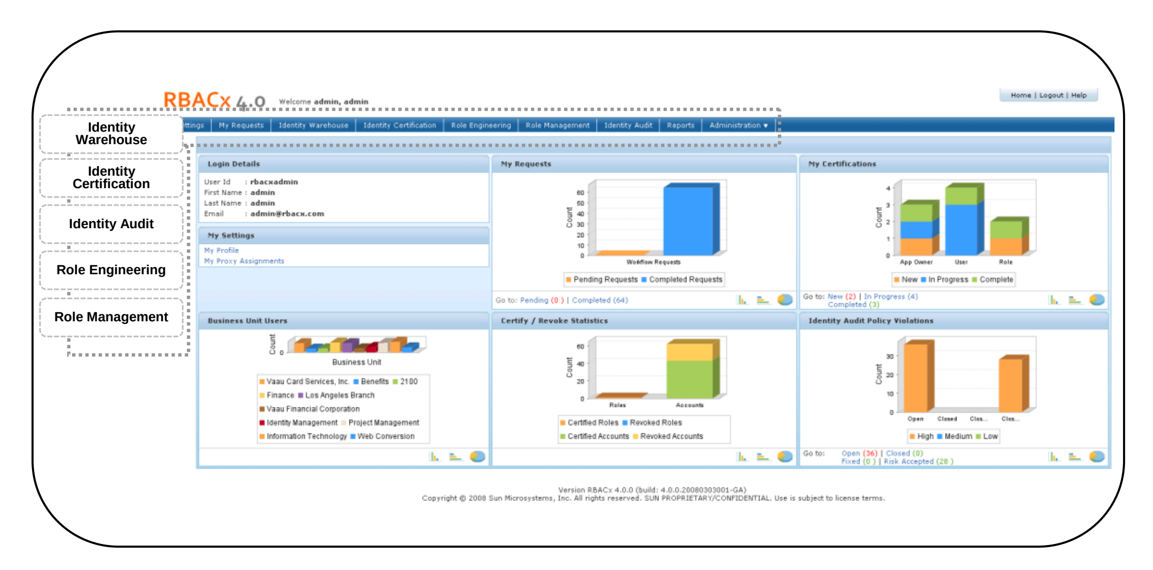


Adapted from [Vaau06a], page 1

Information 29: State of the Art – Sun Role Manager Component Relationship

Dieses Diagramm zeigt die Datentypen *business unit*, *business user*, *role*, *policy*, *user store* und *application*, *end point*, *namespace*, *attribute category* und *attribute* sowie die Beziehungen zwischen diesen Typen. Eine Geschäftseinheit (engl. *business unit*) stellt eine organisatorische Struktur in einem Unternehmen, wie etwa eine Abteilung oder ein Team dar. Geschäftseinheiten sind in der Regel ineinander geschachtelt, um die Hierarchie der Geschäftsstruktur zu repräsentieren. Zugleich ist diese Einheit von zentraler Bedeutung, weil sämtliche Operationen in den drei Aufgabenbereichen von SRM auf Geschäftseinheiten arbeiten. Ein Unternehmensbenutzer (engl. *business user*) ist eine diskrete Einheit innerhalb eines Unternehmens, die bei der Ausübung ihrer Rechte den Informationsbestand des Unternehmens bearbeiten und verändern kann. In der Regel ist dies ein menschlicher Benutzer, es kann sich dabei aber auch um einen Dienst oder ein Programm handeln.

Ein Unternehmensbenutzer kann in diesem Komponentenmodell zu einer oder beliebig vielen Geschäftseinheiten gehören, soweit dies seine Rechte ermöglichen. Im Sinne der Hierarchiebildung verfügt er über genau einen weiteren Unternehmensbenutzer, der über ihm angeordnet ist und der über die Rechte verfügt, seine Rollenzuweisungen zu verwalten und zu überprüfen [Vaau07a, Vaau07b]. Anders als in der Definition in dieser Arbeit wird beim SRM der Unterschied einer Geschäftsrolle zu einer Systemrolle nicht explizit vorgenommen, obwohl das Werkzeug beide Prinzipien kennt. In der Nomenklatur des SRM steht der Begriff „Rolle“ für eine geschäftliche Aufgabe, wohingegen die Einheit zum Kapseln von technischen Zugriffsrechten „Policy“ genannt wird. Dennoch entsprechen diese Einheiten ihrer Konzeption nach den Geschäfts- und Systemrollen. Der Role Manager bietet sowohl bei Geschäftseinheiten als auch bei Rollen eine Hierarchiebildung an, nicht aber bei den Systemrollen. Ein Unternehmensbenutzer kann über beliebig viele Rollen verfügen, die ihrerseits eine beliebige Anzahl von Policys aufweisen. Mit dem Benutzerverzeichnis (engl. *user store*) ist diejenige Datenquelle einer oder mehrerer Anwendungen (engl. *applications*) gemeint, aus der das *identity warehouse* die Benutzer in sein Metaverzeichnis importiert. Dort sind demnach die konkreten Benutzerobjekte abgelegt. Die Assoziation mit einer oder mehrerer Policys definiert die technischen Zugriffsrechte eines Benutzers. SRM versteht die technischen Endsysteme, die unter seiner Verwaltung stehen, allgemein als Anwendungen. Betrachtet man die Komponente Anwendung (engl. *application*), so legt das Komponentenmodell des SRM fest, dass sie einen Besitzer (engl. *owner*) oder Verantwortlichen (engl. *approver*) haben kann, der seinerseits dargestellt ist als Unternehmensbenutzer, im eigentlichen Sinne also als Geschäftsrolle. Jede Anwendung wird in SRM als eigener Namensraum (engl. *namespace*) definiert. Da es zu einer Anwendung aber mehrere Verbindungen geben kann, wie beispielsweise dem Zugriff auf unterschiedliche Datenbanken innerhalb eines Datenbankservers, unterstützt der Role Manager mehrere Verbindungsinstanzen (engl. *end point*), die ihrerseits aus einer Menge von Attributen (engl. *attribute*) bestehen. Attribute bezeichnen in diesem Zusammenhang systemspezifische Eigenschaften, wie etwa den Anmeldenamen eines Benutzers oder dessen Gruppenmitgliedschaften in einem dieser Systeme. Für eine vereinfachte Darstellung ist es möglich, diese Attribute zu Attributkategorien (engl. *attribute category*) zu gruppieren.

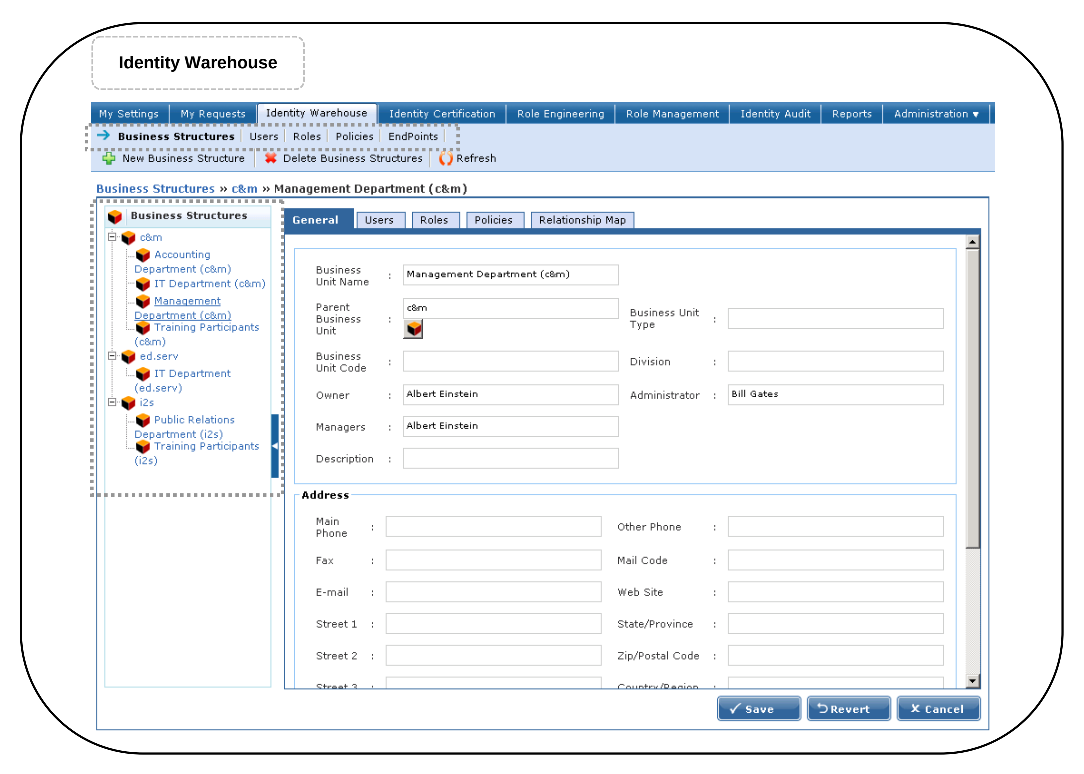


Information 30: State of the Art – Sun Role Manager GUI

Abschließend soll hier ein kurzer Einblick in die grafische Bedienoberfläche des Sun Role Manager gegeben werden, weil in den folgenden Kapiteln die einzelnen Funktionen in Bezug zu dieser grafischen Oberfläche vorgestellt werden. Information 30 zeigt die Startseite nach der Benutzeranmeldung in der aktuellen Version 4.0. Sie bietet eine Übersicht über die einzelnen Arbeitsbereiche des Role Manager. Über die Menüleiste sind die einzelnen Bereiche anzusteuern. Hier wird unmittelbar der Zusammenhang zur Gesamtarchitektur aus Information 28 deutlich. Die Anwendung wird über einen Internet-Browser aufgerufen und bietet eine einheitli-

che Oberfläche für alle Aufgaben im Umgang mit Rollen an. Nachdem nun ein Einblick in das Architektur- und Komponentenmodell des SRM gegeben wurde, folgt eine nähere Betrachtung der Funktionen innerhalb der drei Arbeitsbereiche. Da die Basis aller Bereiche die zentrale Datenbank ist, wird im nächsten Teilkapitel zunächst das *identity warehouse* dargestellt.

3.3.2 Die Komponente *identity warehouse*



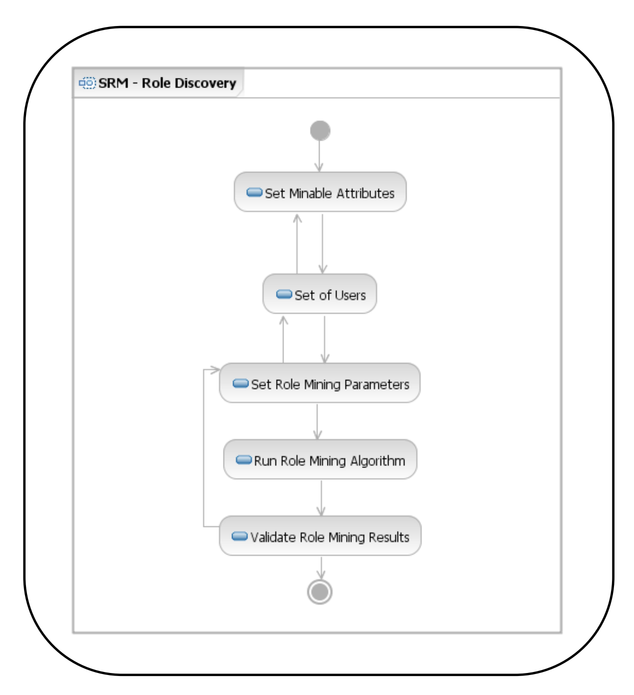
Information 31: State of the Art – Sun Role Manager Identity Warehouse

Diese Abbildung verdeutlicht den Informationsbestand des *identity warehouse*, das den zentralen Datenspeicher im Sun Role Manager darstellt. Wie zunächst ersichtlich ist, sind die Daten, die in dieser Datenbank abgelegt sind, unterteilt in Geschäftsstrukturen (engl. *business structures*), Benutzer (engl. *users*), Rollen (engl. *roles*), Policys (engl. *policys*) sowie Systemschnittstellen (engl. *end points*). Diese Datentypen, zusammen mit den Beziehungen untereinander, wurden im Komponentenmodell in Information 29 beschrieben. In Information 31 ist als Beispiel speziell die Unterkomponente *business structures* innerhalb des Gesamtdatenbestandes abgebildet. Diese Struktur vereint beliebig viele, voneinander unabhängige Geschäftseinheiten unter einem gemeinsamen Wurzelknoten. Jede Geschäftseinheit verfügt über eine Menge an Eigenschaften, wie etwa einem Manager, einer für diese Einheit verantwortlichen Person oder einem Administrator. Zusätzlich können Informationen wie die nächsthöhere Hierarchiestufe oder zentrale Adressdaten hinterlegt werden. Über die anderen Reiter auf der Navigationsleiste erreicht man das Benutzerverzeichnis, das Rollenverzeichnis, den Datenbestand an Policys sowie alle auf dem System vorhandenen Systemschnittstellen. In den folgenden Teilkapiteln wird auf diese Bereiche separat eingegangen, so dass sie an dieser Stelle nicht explizit erwähnt werden.

3.3.3 Die Arbeitsbereiche *role engineering* und *role management*

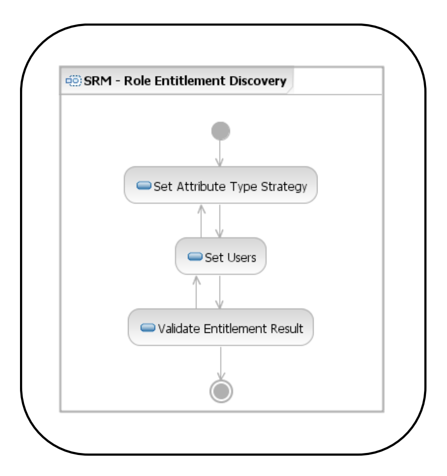
Bei der konkreten Umsetzung von RBAC in einem Unternehmen beginnt man typischerweise damit, ein an das Unternehmen angepasstes Rollenkonzept auf der Basis eines Rollenmodells zu entwickeln, ehe Verwaltungsaufgaben im Bezug darauf anfallen. Deshalb wird zunächst auf die Komponenten *role engineering* und anschließend auf *role management* eingegangen. SRM unterteilt den *role engineering*-Bereich in drei voneinander unabhängige Prozessschritte. Dies sind das Zusammenfassen von Berechtigungen zu Rollen (engl. *role discovery*), das Analysieren von

bestehenden Rollen, Berechtigungen oder Zugriffsmustern (engl. *role entitlement discovery*) und das Erstellen von Regeln, mit deren Hilfe das Eintragen von Benutzern in Rollen eingeschränkt werden kann (engl. *rule discovery*). Diese drei Teilprozesse werden nun beispielhaft gezeigt.



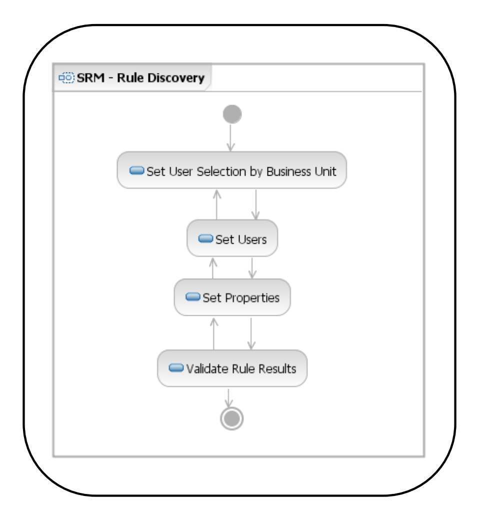
Information 32: State of the Art – Sun Role Manager Role Discovery

Hauptaufgabe der Komponente *role discovery* ist es, Rollen als logische Einheiten für Zugriffsberechtigungen zu finden. Neben dem manuellen Anlegen von Rollen bietet der Sun Role Manager einen teilautomatisierten Prozess zur Herausarbeitung von Rollen aus den bestehenden Daten des Unternehmens an. Hierbei werden Zusammenhänge zwischen Zugriffsberechtigungen unterschiedlicher Benutzer hergestellt und analysiert, um eine möglichst große Schnittmenge zu generieren und daraus schließlich Rollen abzuleiten. Die gemeinsamen Berechtigungen werden in den Rollen dabei in Form von Policys gekapselt, was dem Begriff der Systemrolle in dieser Arbeit entspricht. Dieser gesamte Prozess basiert in SRM auf der Anwendung von *role mining*-Algorithmen, wie bereits in Kapitel 3.1.1 erwähnt wurde. Die Daten, auf denen sie operieren, werden vorab in Einzelschritten vom Anwender ausgewählt. Der gesamte Prozess ist in Information 32 dargestellt und zeigt die einzelnen Aktivitäten vom initialen Auswählen der Attribute bis zur Speicherung neuer Rollen. Um bei der Anwendung der *role mining*-Algorithmen ein möglichst gutes Ergebnis zu erhalten, ist es wichtig, den Datenbestand des *identity warehouse* vor dem Suchlauf des Algorithmus auf den jeweils wesentlichen Ausschnitt einzuschränken. Dazu wählt man für jedes einzelne Unternehmenssystem, das im speziellen Fall von Interesse ist, eine Menge von Attributen aus, auf denen der Algorithmus operieren soll. Anschließend sollte die Menge der zu untersuchenden Benutzer eingeschränkt werden. Hier bietet SRM die Möglichkeit, Benutzer nach der Zugehörigkeit zu einer Geschäftseinheit, zu einem Unternehmenssystem, zu einer bereits bestehenden Rolle, oder global auszuwählen. In der nächsten Aktion lassen sich die Parameter des Algorithmus spezifizieren, um das Ergebnis des Suchlaufs zu steuern. Zu diesem Zeitpunkt verfügt SRM über alle Daten, die zur Berechnung nötig sind. Nach Beendigung der Berechnung werden die Ergebnisse ausgegeben, die anschließend noch verfeinert werden können. Der gesamte Prozess ist iterativ und kann somit schrittweise an die gewünschte Granularität angepasst und individuell optimiert werden. Das Ergebnis kann dann in einem abschließenden Schritt als Rolle, die ihrerseits über Policys verfügt, im *identity warehouse* des SRM abgelegt werden.



Information 33: State of the Art – Sun Role Manager Role Entitlement Discovery

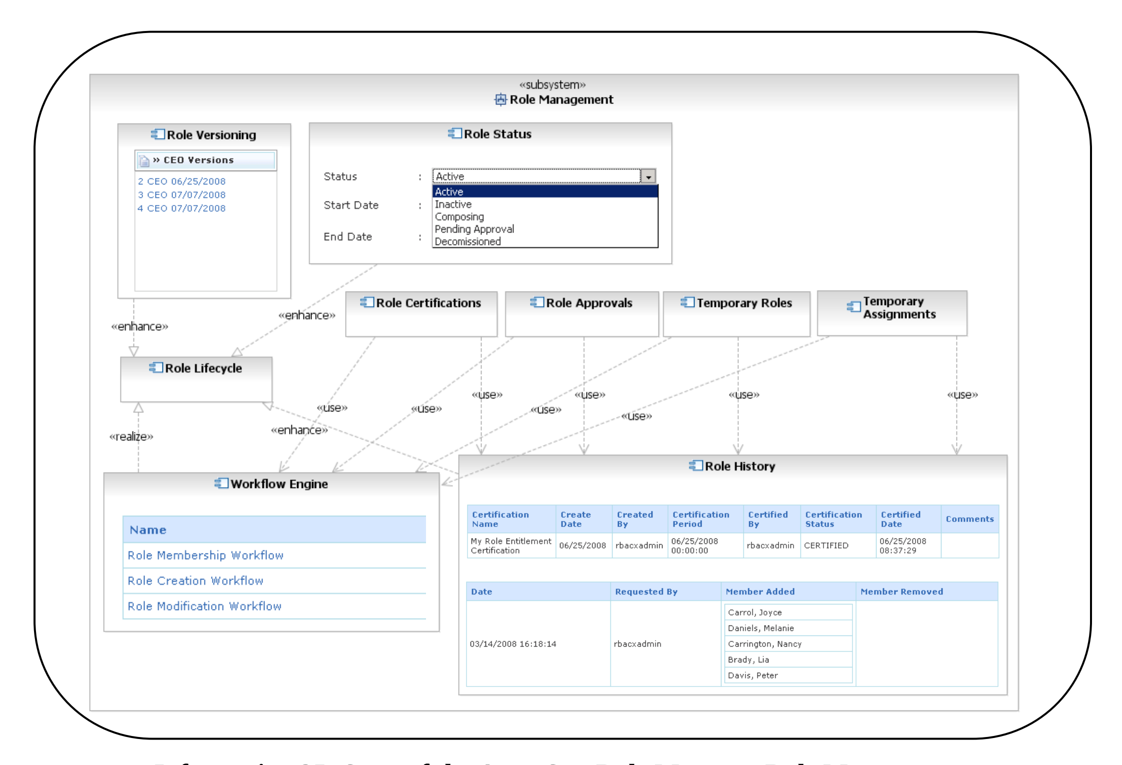
Nachdem im ersten Teilprozess neue Rollen und Policys definiert wurden, beschäftigt sich *role entitlement discovery* mit der Überarbeitung bereits definierter Rollen zu einem späteren Zeitpunkt im Lebenszyklus, wobei auch hier *role mining*-Algorithmen zum Einsatz kommen. Information 33 stellt diesen Prozess als Aktivitätsdiagramm dar. Im Gegensatz zu *role discovery* sind hier Rollen bereits definiert, so dass die Berechnung hier nicht mehr auf der Basis von Benutzern, sondern bereits auf der Basis von Rollen stattfindet. Zunächst beginnt man allerdings damit, eine Strategie zur Attributauswahl zu bestimmen. Der Sun Role Manager bietet als Strategie einerseits die Auswertung aller Attribute in den Policys der selektierten Rollen an, womit erreicht werden kann, dass systemübergreifende Policys entwickelt werden können. Andererseits verfolgt die zweite Strategie, bei der nur gewisse SRM-internen Attribute zur Rolleneinteilung (engl. *entitlement attributes*) ausgewählt werden können, das Ziel, Rollen zu konsolidieren. Die Konsolidierung der bereits definierten Rollen wird dadurch erzielt, dass lediglich die Attribute der SRM-eigenen Datentypen analysiert werden, nicht aber diejenigen der technischen Endsysteme. Im nächsten Schritt können die Benutzer aus allen bekannten Rollen bestimmt werden, für die die Analyse vorgenommen werden soll,. Durch dieses Vorgehen können alle Benutzer in die Analyse mit einbegriffen werden, die über mindestens eine Rolle verfügen. Die Ergebnisliste schließlich lässt sich so filtern, dass von allen gefundenen Attributen nur noch diejenigen enthalten sind, über die mindestens ein gewisser Prozentsatz aller Benutzer verfügt. Somit wird gewährleistet, dass die beste Überdeckung für die ausgewählten Benutzer erfasst wird. In einem letzten Schritt kann diese Zusammenstellung in Form einer oder mehrerer Policys im *identity warehouse* abgelegt werden, die dann ihrerseits automatisch mit den ausgewählten Rollen verknüpft werden.



Information 34: State of the Art – Sun Role Manager Rule Discovery

Der dritte Prozess *rule discovery* hat zum Ziel, die Einteilung von Benutzern in Rollen auf der Basis von vordefinierten Benutzerattributen in SRM durch das Definieren von Regeln einzuschränken. Dazu werden Klassifizierungsregeln für Rolle verwendet, die durch die Anwendung von *role mining*-Algorithmen erstellt werden. Dieser Prozess, der in Information 34 dargestellt ist, beginnt mit der Auswahl von Benutzern als Eingabeparameter für den Algorithmus. Im Anschluss daran können sowohl die zur Verfügung stehenden Benutzerattribute, als auch die Parameter des Algorithmus selbst spezifiziert werden, um den Suchlauf zu verfeinern. Im anschließenden Suchlauf wird versucht, gemeinsame Attributwerte bei den ausgewählten Benutzern zu entdecken, um diese als Klassifizierungsregeln im *identity warehouse* abzulegen und mit den Rollen zu verknüpfen. Um in die Rolle aufgenommen werden zu können, die mit der Klassifizierungsregel belegt ist, muss ein Benutzer daher über die in der Regel festgelegte Bedingung verfügen. Hier zeigt sich der Zusammenhang zwischen Policys und Regeln: Beide Prinzipien arbeiten auf Attributebene, aber Policys schränken die Zugriffsrechte der Rollenmitglieder ein, während Regeln die Rollenmitgliedschaft selbst einschränken. Policys werden demnach für Attribute der Unternehmenssysteme effektiv, wohingegen Regeln die Benutzerattribute in SRM einschränken, die notwendig sind, um einer Rolle zugewiesen werden zu können.

Diese drei geschilderten Prozesse stehen im Sun Role Manager zur Verfügung, um Geschäfts- und Systemrollen zu definieren, abzuändern und miteinander in Verbindung zu bringen (engl. *role engineering*). Im Folgenden werden die Aufgaben im Zusammenhang mit der Verwaltung von Rollen (engl. *role management*) beschrieben. Information 35 zeigt diese Aufgabenbereiche schematisch und setzt sie zueinander in Bezug. Damit soll der Zusammenhang zur Komponentenarchitektur aus Information 28 noch einmal verdeutlicht werden.



Information 35: State of the Art – Sun Role Manager Role Management

Ein zentraler Begriff im Rollenmanagement des SRM ist der Lebenszyklus von Rollen. Dieser beschäftigt sich mit der Tatsache, dass eine Rolle mehrere Phasen durchläuft, von der initialen Definition, dem Hinzufügen von Systemrollen, den Änderungen im Laufe der Zeit sowie den Bewilligungen hierfür. Der Lebenszyklus einer Rolle endet mit Maßnahmen zu ihrer planmäßigen Deprovisionierung. Dieser Lebenszyklus wird technisch durch die Komponente unterstützt, die die internen *workflows* ausführt (engl *workflow engine*). Hierbei existieren drei vordefinierte Arbeitsabläufe im Zusammenhang mit dem Lebenszyklus. Zunächst gibt es einen *workflow*, der

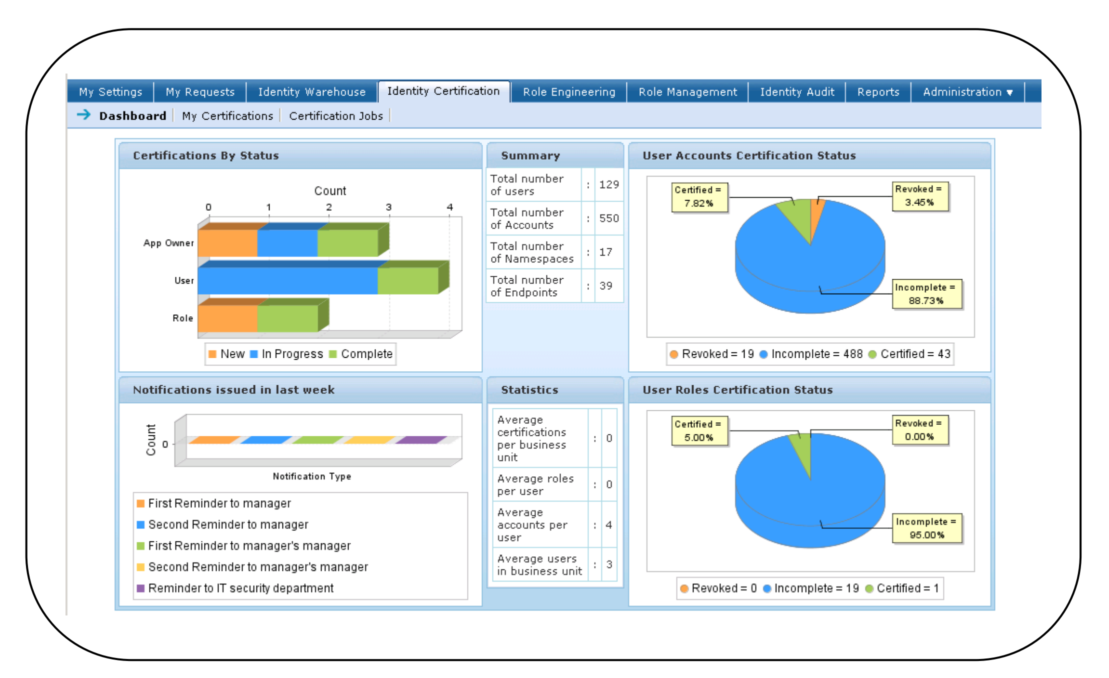
sich mit der Bewilligung von Änderungen an Rollenmitgliedschaften beschäftigt. Ein zweiter steuert den Ablauf bei der Erzeugung einer Rolle sowie den darin enthaltenen Policys und ein dritter regelt Änderungen an den Rollen sowie den Policys selbst. Alle Aufgaben des Rollenmanagements verwenden diese drei Arbeitsabläufe, wobei sie an gewissen Stellen angepasst werden können. So ist es etwa möglich, Teilaufgaben an die dafür verantwortliche Rolle weiterleiten zu lassen oder automatische Emails auf der Basis vorgefertigter Vorlagen zu verschicken. Diese Abläufe gewährleisten ein geplantes und zielorientiertes Arbeiten im Umgang mit Rollen.

Ein zweiter Aspekt, der unmittelbar mit den *workflows* zur Realisierung des Lebenszyklus zu tun hat, ist der Status einer Rolle. Hierbei gibt es fünf vordefinierte Stati, in der sich eine Rolle befinden kann. Zusätzlich zum aktuellen Status kann ein Zeitintervall angegeben werden, für den eine Rolle aktiv ist, um sie am Ende ihres Lebenszyklus durch einen automatisch initiierten *workflow* deprovisionieren zu lassen. Dies stellt sicher, dass keine Unternehmensbenutzer länger Zugriff auf die internen Ressourcen der Abteilung oder des gesamten Unternehmens hat, als nötig. Dieser Status drückt den momentanen Zustand der Rolle im Rollenmanagement des SRM aus. So ist eine neu definierte Rolle, die im Zuge eines *workflows* angelegt wurde erst zu dem Zeitpunkt aktiv, zu dem sie durch den Rollenverantwortlichen freigeschaltet wurde, oder diese Aufgabe nach Ablauf einer Frist an jemand anderen weitergeleitet wurde, der die Freischaltung durchführt. SRM bietet hierfür eine Integration in die grafische Bedienoberfläche an, so dass jede Rolle über die von ihr bereits erledigten oder noch nicht vervollständigten Arbeitsaufgaben informiert ist.

Ein letzter Aspekt bezüglich des Lebenszyklus stellt ein Protokollmechanismus für Änderungen an Rollen dar. Durch diesen Mechanismus werden im Rahmen des Rollenmanagements Planungen in der Zukunft sowie Analysen von vergangenen Änderungen ermöglicht. Dazu stellt SRM Historientabellen für Rollen in unterschiedlichen Ausprägungen bereit. Einerseits protokolliert der Role Manager, wann welcher Benutzer zu einer Rolle hinzugefügt oder aus ihr entfernt wurde sowie wer diesen Schritt durchführte. Eine andere Sicht auf den Zustand einer Rolle bringt die Versionierung mit sich. Hierbei werden Änderungen an Rollen nicht überschrieben, sondern stattdessen eine neue Version der Rolle erzeugt und der bisherige Stand zusätzlich in der Rolle mitgeführt. Dies geschieht bei jeder Änderung an Rolleneigenschaften, um eine lückenlose Historie zu gewährleisten.

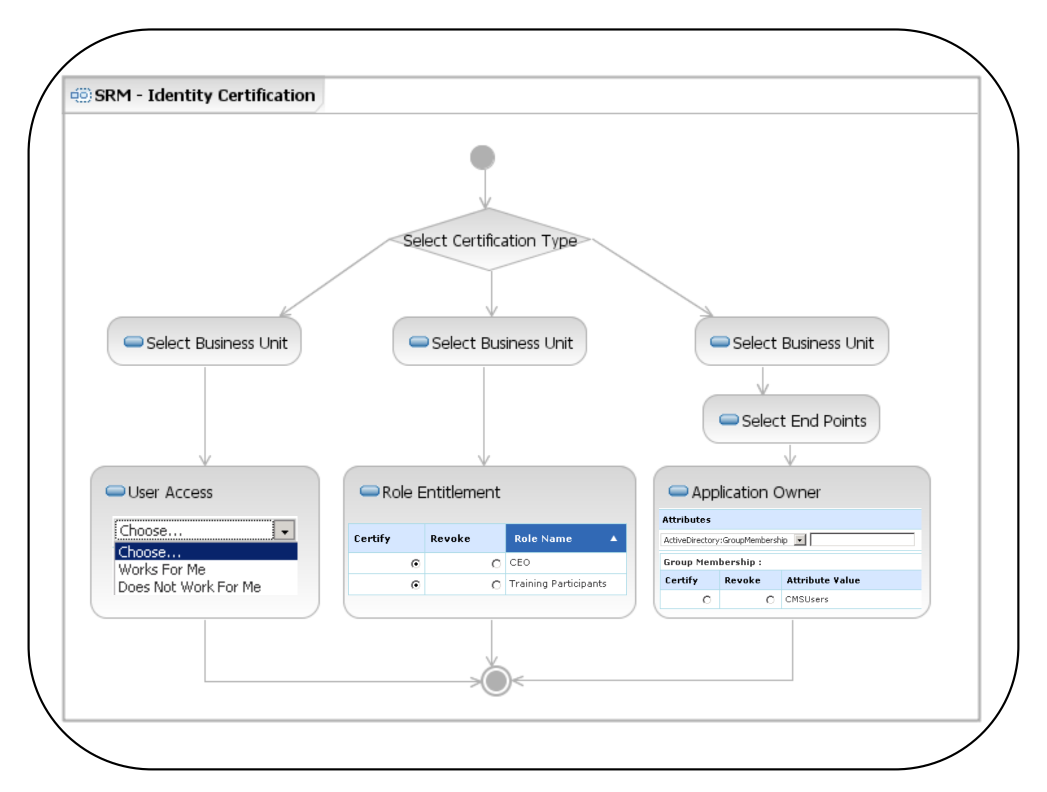
3.3.4 Der Arbeitsbereich *identity certification*

Der Arbeitsbereich *identity certification* verkörpert das Verwalten von Zuweisungen jeglicher Ausprägung im Sun Role Manager. Hierbei ist es durch die integrierten Arbeitsabläufe (engl. *workflows*) möglich, den Verwaltungsprozess aufzuteilen und so die Zuweisungen an das jeweils dafür verantwortliche Personal zu delegieren. Diese Aufteilung unterstützt somit das Dezentralisieren von Arbeiten innerhalb eines Unternehmens. Der gesamte Bereich umfasst zwei Teilaufgaben: Einerseits das Überwachen des aktuellen Standes der Zuweisungen im Unternehmen und andererseits das Durchführen dieser Zuweisungen (engl. *certifications*). Abgeleitet von der Übersetzung aus dem Englischen wird dafür auch der Begriff „Zertifizierungen“ zur Bezeichnung von Zuweisungen verwendet.



Information 36: State of the Art – Sun Role Manager Identity Certification Dashboard

Information 36 zeigt die grafische Benutzeroberfläche für einen Rollenverantwortlichen. Eine Rolle kann in SRM unterschiedlichen Arten von Zuweisungen besitzen: Zum Einen gibt es die Zuweisung von Benutzern zu Geschäftseinheiten, etwa einer Abteilung, zum Zweiten die Zuweisung von Benutzern zu Geschäftsrollen und zum Dritten die Zuweisung von Systemrollen zu Geschäftsrollen. Diese Aufteilung ermöglicht es, die Arbeitsprozesse sehr feingranular zu gestalten. Genauso ist es aber möglich, dass ein und dieselbe Person alle Aufgaben selbst erledigt. Die Arbeitsabläufe, die diese drei verschiedenen Zertifizierungen als Prozess realisieren, basieren dabei auf einer *workflow engine* zusammen mit einer Entwicklungskomponente für *workflows*, die über eine grafische Bedienoberfläche verfügt, mit der *workflows* gestaltet und angepasst werden können.

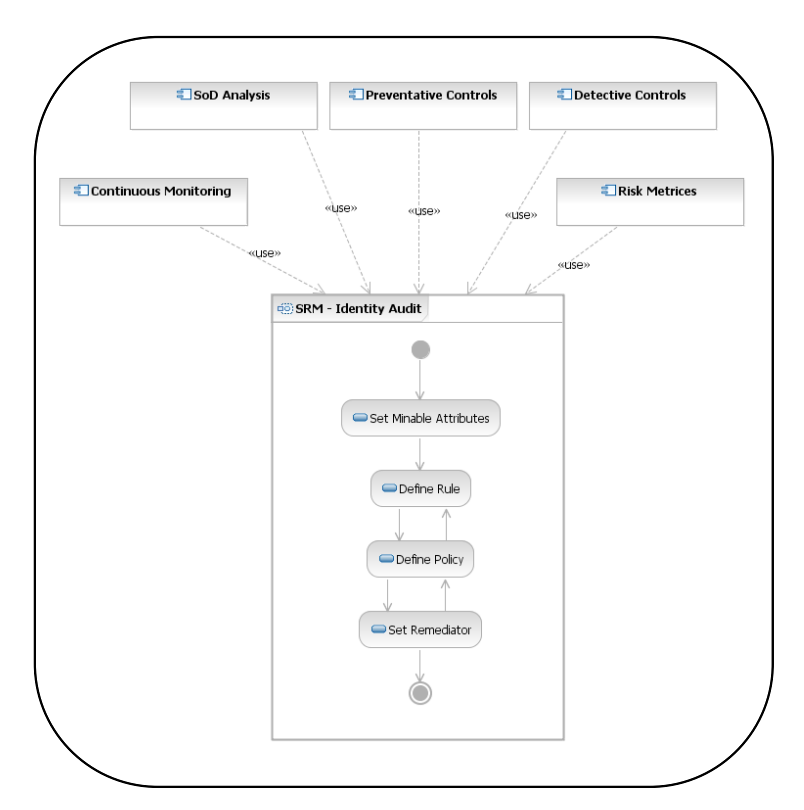


Information 37: State of the Art – Sun Role Manager Identity Certification

Information 37 stellt die drei verschiedenen Zertifizierungstypen grafisch dar. Eine wichtige Aufgabe im Rollenmanagement ist es, regelmäßig zu überprüfen, ob die Benutzer noch aktiv sind, die Abteilung gewechselt haben oder auch aus der Firma ausgeschieden sind. SRM stellt hierfür den Zertifizierungstyp *user access* bereit. Dieser arbeitet auf Basis von Geschäftseinheiten und richtet sich an den Verantwortlichen dieser Geschäftseinheit. Für jeden Benutzer muss vom Verantwortlichen angegeben werden, ob er der Abteilung noch angehört, oder nicht. Dieser Schritt kann dadurch teilautomatisiert werden, dass der Prozess an einem vorher fest definierten Tag automatisch startet und den Verantwortlichen an die Überprüfung erinnert. Der zweite Zertifizierungstyp *role entitlement* richtet sich an den Rollenverantwortlichen innerhalb des Unternehmens. Seine Aufgabe ist es, den Umfang einer Rolle abzuändern. Dazu wählt er zunächst eine Geschäftseinheit aus und selektiert die zu betrachtenden Rollen. Im Anschluss daran können die Policys, die in den Rollen gekapselt sind sowie alle Attribute innerhalb der Policys einzeln zertifiziert werden. Der dritte Zertifizierungstyp *application owner* richtet sich an das Personal, welches sich um die technische Pflege der Unternehmenssysteme kümmert und in diesem Sinne als Besitzer der Anwendungen angesehen werden kann. Sie können die konkreten Attributwerte für die in den Policys definierten Attribute festlegen. Insgesamt betrachtet zeigt sich hier, wie SRM den technischen Wissenstand unterschiedlichen Personals unterstützt: Während sich *user access certification* mit der Verwaltung der Mitglieder von Geschäftsrollen befasst, verlangt *role entitlement certification* bereits technisches Detailwissen, da hier die Verknüpfung von Geschäfts- und Systemrolle zum Tragen kommt. Dies drückt sich in SRM in Form von Policys aus, was als Systemrolle verstanden werden muss. Diese kapseln technische Spezifika der Unternehmenssysteme in sich als Attribut. *Application owner certification* schließlich definiert die konkreten Variablenwerte für die eben erwähnten Attributwerte und richtet sich damit klar an technisch sehr versiertes Personal.

3.3.5 Der Arbeitsbereich *identity audit*

Die Überwachung von Ausnahmen und Regelverletzungen ist Hauptbestandteil des Arbeitsbereichs *identity auditing and management*. Der Role Manager bietet einen einheitlichen Prozess an, mit dessen Hilfe die vielfältigen Ursachen für Verletzungen erkannt und in organisierter Weise behandelt werden können. Dieser Sachverhalt ist in Information 38 dargestellt.



Information 38: State of the Art – Sun Role Manager Identity Audit

Die erfassbaren Regelverletzungen operieren in SRM auf Attributebene. Dadurch ist es möglich, sowohl auf Attribute von Geschäftsrollen, aber auch auf die von Systemrollen zuzugreifen. In diesem Zusammenhang werden Verletzungen definiert und diese anschließend überwacht. Durch die Attribute der Geschäftsrollen ist es möglich, SoD-Verletzungen zu erkennen. Um feingranulare Regelverletzungen auf Systemebene erkennen zu können, werden die Attribute der Unternehmenssysteme und deren Wertebelegung zur Überwachung hinzugefügt. Dies geschieht über die bereits erwähnten Regeln (engl. *rules*), die aus einer Menge von Attribut/Wert Paaren bestehen und somit konkrete Bedingungen darstellen. Mehrere Regeln können in einer Überwachungs-Policy zusammengefasst und durch logische Operatoren in Beziehung zueinander gesetzt werden. Anschließend wird eine Person bestimmt, die im Falle eines Verstoßes gegen die definierte Bedingung informiert wird. Zu diesem Zeitpunkt ist die Policy korrekt definiert und aktiv. Sie reagiert im Falle einer Verletzung und wendet sich in diesem Fall an die in der Policy definierte verantwortliche Person. Darüber hinaus lässt sich in einer Policy ein bestimmter Zeitpunkt angeben, an dem sie aktiv werden soll um auf das *identity warehouse* zuzugreifen und den Datenbestand bzw. die spezifizierte Geschäftseinheit zu analysieren.

Die Überwachung beginnt dabei bei der initialen Definition, setzt sich mit der Überprüfung nach Verletzungen fort und endet mit dem Auftreten der Verletzung und deren Behandlung. Bei der Behandlung hat man in SRM durch einen menschlichen Akteur die Möglichkeit, die Verletzung zu ignorieren, sie zu behandeln oder an einen anderen Verantwortlichen weiterzuleiten. Dabei werden alle Vorkommnisse protokolliert, so dass die erfolgten Behandlungen zu einem späteren Zeitpunkt nochmals eingesehen und ausgewertet werden können. Die dabei protokollierten Daten umfassen sowohl das Datum der Verletzung, also auch das Datum der Fehlerbehandlung und den Bearbeiter.

3.4 Resümee

In diesem Kapitel wurde der aktuelle Stand bei Rollenmodellen aus der Forschung und Rollenmanagementansätze aus dem Bereich der Technik beschrieben.

Bei den wissenschaftlichen Arbeiten wurde zunächst ein Modell zur automatischen Entwicklung von Rollen vorgestellt, welches Mechanismen aus dem *data mining* auf Rollen überträgt. Dieser Ansatz bietet sich an, weil die Informationen, die zu Rollen und Berechtigungen führen, in Datenbeständen vorhanden sind. Wie aus Implementierungen hervorgeht, lässt sich dadurch eine beträchtliche Verringerung der Zeit erreichen, die zur Entwicklung von Rollen benötigt wird. Danach wurde ein Rollenmodell vorgestellt, das Rollen für die unternehmensweite Zugriffskontrolle modelliert. Technische Umsetzungen dieses Rollenmodells für Unternehmen (ERBAC) haben allerdings gezeigt, dass dieses Modell zu einer Vielzahl von Rollen führt, so dass Erweiterungen vorgeschlagen wurde, um die Komplexität von ERBAC in technischen Umsetzungen zu verringern. Diese Erweiterungen wurden ebenso dargestellt, wie der Mehrwert, der sich potentiell aus ihnen ergibt. Ein weiteres Modell befasste sich mit dem Lebenszyklus von Rollen und definierte das Rollenmanagement als einen ganzheitlichen Prozess, der Veränderungen beachtet, denen ein implementiertes Rollenmodell im Wirkbetrieb unterliegt. Dazu wurde in Anlehnung an die klassische Software-Entwicklung ein Lebenszyklus für Rollen eingeführt. Der Vorteil davon ist, dass das Entwickeln von Rollen als ganzheitlicher Prozess betrachtet wird, der nicht mit dem entwickelten Rollenmodell endet, sondern die Pflege und Nachbesserung der Rollen mit einbezieht.

Im Bereich der Industrie sind in den letzten Jahren Rollenmanagementwerkzeuge entwickelt worden, die als zentrale Steuereinheit für die Zugriffskontrolle angesehen werden können. Mit dem Omada Identity Manager und dem Sun Role Manager wurden zwei aktuelle Werkzeuge betrachtet und dabei zunächst ein Überblick über deren Gesamtarchitekturen und verwendeten Datentypen gegeben. Im Anschluss daran folgte jeweils ein detaillierter Einblick in die Aufgabenbereiche, die die beiden Werkzeuge im Bereich des Rollenmanagements erfüllen. Dabei wurden diese Bereiche jeweils in Bezug gesetzt zu den typischen Aufgaben von Rollenmanagementwerkzeugen aus Kapitel 2.3. Dabei wurde das Ziel verfolgt, die Vorstellung der Produkte wertungsfrei zu halten, um einen objektiven Überblick zu gewährleisten. Auf die Bewertung wird in Kapitel 6 dieser Arbeit dediziert eingegangen und dafür zunächst ein Kriterienkatalog entwickelt, der anschließend auf beide Werkzeuge angewandt wird.

Wie in diesem Kapitel dargestellt ist, weisen aktuelle Rollenmodelle, die für den Einsatz in Unternehmen konzipiert wurden, zwei Schwachstellen auf: Einerseits vermischen sich geschäftliche und technische Aspekte und andererseits führen sie zu komplexen technischen Implementierungen in der praktischen Umsetzung. Auf der Basis der in diesem Kapitel dargestellten Erkenntnisse wird im nun folgenden Kapitel ein Rollenmodell entwickelt, welches sich diesen beiden Schwachstellen widmet.

4 ENTWICKLUNG EINES ROLLENMODELLS ZUR ABBILDUNG VON GESCHÄFTS- UND SYSTEMROLLEN

4.1 Anforderungen an das Rollenmodell

Dieses Kapitel stellt den ersten zentralen Aspekt dieser Arbeit dar. Es wird ein Modell für die rollenbasierte Zugriffskontrolle entwickelt, das auf dem RBAC Standardrahmenwerk basiert, welches in Kapitel 2.2 bereits beschrieben wurde. Dieses Modell unterteilt das Element „Rolle“ in die zwei Ausprägungen „Geschäftsrolle“ (engl. *business role,* BR) und „Systemrolle (engl. *system role,* SR) und richtet sich mit dieser Unterscheidung speziell an Unternehmen. Im Folgenden wird dieses Modell daher auch als BRBAC (engl. *business role-based access control*) bezeichnet. Dazu werden in diesem Unterkapitel zunächst zwei Ziele Z_1 und Z_2 für das Rollenmodell definiert und Anforderungen erhoben, um diese Ziele umzusetzen. In den Kapiteln 4.2 und 4.3 werden die erhobenen Anforderungen im Bezug zu dem Ziel modelliert, welches sie erfüllen und detailliert beschrieben. Kapitel 4.4 stellt dann das resultierende Gesamtbild des Rollenmodells BRBAC vor. An dieser Stelle sei ergänzend erwähnt, dass sich im Kontext der rollenbasierten Zugriffskontrolle verschiedene Bezeichnungen für ein und denselben Sachverhalt durchgesetzt haben. Um Missverständnisse zu vermeiden, werden diese Begriffe nun zueinander in Bezug gesetzt. Ein wichtiger Aspekt bei Zugriffskontrollarchitekturen stellt die Spezifikation des Zugriffs dar. Hierfür haben sich die Begriffe *permission* oder „Policy“ durchgesetzt. Sowohl in RBAC, wie auch in ERBAC wird eine Berechtigung allgemein als *permission* bezeichnet, wie aus den Kapiteln 2.2 und 3.1.2 hervorgeht. Diese Bezeichnung unterstreicht die Nähe zu technischen Systemen, in denen Berechtigungen auch als *permission* definiert sind. Auf der Geschäftsebene hat sich zur Formulierung von Berechtigungen hingegen der Begriff „Policy“ etabliert. Diese Arbeit verwendet hingegen in Anlehnung an [Em08] den Begriff „Policy“ als Bezeichnung für Berechtigungen und unterscheidet dabei durch ein geeignetes Präfix, ob sie aus Geschäftszielen abgeleitet wurden, oder zur Unterstützung von technischen Zugriffsrechten in Systemen dienen. Im Folgenden wird zur Unterscheidung dieser beider Bereiche übereinstimmend von Geschäfts- oder System-Policys gesprochen.

Aktuell lässt sich feststellen, dass die Forschung im Bereich RBAC schon sehr weit vorangeschritten ist, jedoch stets von einem sehr allgemeinen Rollenbegriff ausgehen. Die Umsetzungen dieser Modelle in der Industrie weisen jedoch auch aufgrund der fehlenden Unterscheidung unterschiedlicher Rollenbegriffe Schwachstellen auf, die gerade in heterogenen Systemumgebungen zum Tragen kommen, wie Kapitel 3.1.3 bereits aufzeigt. Die beiden Ziele des in diesem Kapitel entwickelten Rollenmodells BRBAC sind, sich einerseits einer differenzierten Betrachtung von Rollen und andererseits der gestiegenen Komplexität zu widmen, die durch die Verwendung verteilter Informationssysteme entsteht. Dabei werden in der Folge Mechanismen im Bezug auf Policys modelliert, die in Rollen subsumiert sind.

Ziel Z_1: Trennung von Geschäfts- und Systemsicht

- **Anforderung $A_{1.1}$: Modellierung von Geschäfts- und Systemrollen.** Diese Anforderung spezifiziert eine explizite Trennung von geschäftlichen und technischen Aspekten, was sich in zwei unterschiedlichen Rollenbegriffen manifestiert. Weder in Rollenmodellen wie NIST-RBAC oder ERBAC, noch in technischen Umsetzungen wird dieser Sachverhalt bisher explizit betrachtet. Auf die Motivation für diese Unterscheidung wird im Folgenden eingegangen.

  Gerade in komplexen Systemumgebungen, wie man sie in Unternehmen typischerweise antrifft, basieren nahezu alle geschäftlichen Prozesse auf Computersystemen, oder werden durch sie unterstützt. Diese starke Zunahme an computergestützter Arbeitsweise führte in den letzten Jahren zu einer ebenso starken Zunahme an Heterogenität des Unternehmensnetzwerks, was sich auch auf die verwendeten Zugriffskontrollarchitekturen

auswirkte. Auf Ebene der technischen Endsysteme existiert ebenfalls ein Rollenbegriff, der sich von der geschäftlichen Sichtweise jedoch stark unterscheidet. Wie bereits angesprochen wurde, ist eine klare Tendenz hin zu rollenbasierten Zugriffskontrollansätzen zu erkennen. In vielen heutigen Rollenmodellen, wie etwa dem ERBAC-Modell (vgl. Kapitel 3.1.2) findet sich ein Rollenbegriff, der sehr stark an Geschäftsfunktionen ausgerichtet ist. Die Auffassung im ERBAC-Modell ist allerdings nicht hinreichend spezifiziert für die Situation in heterogenen Systemlandschaften, da die Komplexität ja insbesondere durch die Vielzahl an technischen Systemen entsteht. Information 1 verdeutlicht diesen Sachverhalt. Diese Schwachstelle wurde im Kreis der Wissenschaft bereits erkannt und insofern nachgebessert, dass eine Rolle neben geschäftlichen Funktionen ebenso technische Zugriffskontrollinformationen verinnerlicht. Diese Modellierung führt jedoch zu einer Mehrdeutigkeit beim Rollenbegriff, der nunmehr nicht klar zwischen diesen beiden Rollendefinitionen unterscheidet (siehe dazu Kapitel 3.1.3). In heterogenen Systemumgebungen, wie man sie heutzutage in Unternehmensnetzwerken findet, existieren diese beiden unterschiedliche Auffassungen des Begriffs „Rolle“, wenn nicht in expliziter, dann zumindest in impliziter Form. Dabei verkörpert eine Rolle einerseits geschäftsnahe Aspekte wie etwa eine geschäftsnahe Funktion. Somit ist bisher weder eine klare Trennung dieser Aspekte gegeben, noch wird die Verbindung zwischen ihnen modelliert. Als Beispiele für geschäftsnahe Aspekte einer Rolle seien etwa ein Jobtitel, eine Anwendungsdomäne – etwa eine Zweigstelle – oder ein Verweis auf einen Vorgesetzten genannt. Andererseits verkörpert der Rollenbegriff technische Zugriffsrechte, die in den vielfältigen Endsystemen auf ganz unterschiedliche Weise dargestellt werden. Die Forderung nach der expliziten Trennung von geschäftlichen und technischen Belangen, die sich in unterschiedlichen Auffassungen des Begriffs „Rolle“ manifestiert, stellt die zentrale Anforderung zur Realisierung von Z_1 in BRBAC dar. Die Anforderung $A_{1,1}$ soll diesen Unterschied formal fassen und somit die Mehrdeutigkeit des Begriffs „Rolle“ beseitigen.

- **Anforderung $A_{1.2}$: Unterstützung von generischen Rollen.** Als Ergänzung zur Anforderung $A_{1,1}$ und in Übereinstimmung zum ERBAC-Modell soll das hier entwickelte Rollenmodell generische Rollen als Mechanismus zur Zusammenfassung ähnlicher Rollen unterstützen. Im Gegensatz zur Generik, wie sie im ERBAC-Modell verwendet wird, soll im BRBAC-Modell zusätzlich das technische Wissen gesenkt werden, das zur Verwaltung des Rollenmodells im Wirkbetrieb vorhanden sein muss. Nach der Formulierung der Anforderung und der Abgrenzung zu bisherigen Modellen wird diese Anforderung nun motiviert.

Aktuell zeigt sich ein reges Interesse an rollenbasierten Zugriffskontrollarchitekturen in Wissenschaft und Technik, was anhand der Publikationszahlen für RBAC ersichtlich ist. Rollenmodelle, die aktuell im Entstehen begriffen sind, sollten somit Anforderungen aus beiden Betätigungsfeldern gerecht werden. Hieraus ergibt sich die Forderung nach einer formalen Spezifikation von Rollenmodellen für den Bereich der Wissenschaft und der Praxisrelevanz für die Technik. Gerade der letztgenannten Forderung wurde bisher nicht in ausreichendem Maße entsprochen, wie Kapitel 3.1.3 bereits belegt. In diesem Kapitel wurde eine Erweiterung des ERBAC-Modells aus Kapitel 3.1.2 vorgestellt, das sich damit auseinandersetzt, dass ERBAC in der praktischen Umsetzung zu einer großen Zahl von Rollen führt und somit der ursprünglichen Forderung nach Verringerung von Komplexität entgegenwirkt. Der daraus resultierenden Unzufriedenheit mit Rollemodellen in der praktischen Umsetzung soll sich das hier vorgestellte Rollenmodell stellen. Gerade im Hinblick auf die Akzeptanz einer Rollenmanagementlösung im Wirkbetrieb wurde das Rollenmodell BRBAC so entworfen, dass es sich positiv auf die Akzeptanz im Unternehmen auswirkt. An einer Rollenmanagementlösung arbeiten für gewöhnlich Mitarbeiter unterschiedlicher technischer Qualifikation. Der Großteil davon ist technisch nicht versiert genug, um das System in seiner Gesamtheit verstehen zu können. Das Rollenmodell sollte dieser Tatsache Rechnung tragen und versuchen, von

technischen Details soweit zu abstrahieren, wie im gegebenen Kontext nötig. Um dies leisten zu können, wird der in $A_{1.1}$ geforderte Rollenbegriff möglichst allgemein – oder generisch – formuliert, um in unterschiedlichen technischen Kontexten gleichermaßen angewandt werden können. Dazu bedient sich das Modell generischer Rollen, wie sie auch in den Erweiterungen zum ERBAC-Modell [Ke02] vorgeschlagen werden. Im Gegensatz zu dem dort verfolgten Ansatz steht hier aber nicht nur die Zusammenfassung ähnlicher Rollen in Endsystemen im Vordergrund, sondern zusätzlich auch die Senkung des nötigen technischen Wissens. Um dies durch ein Beispiel zu verdeutlichen, sei ein RBAC-fähiges Content-Management-System (CMS) in mehreren Abteilungen eines Unternehmens eingesetzt. Da diesen Systemen dasselbe Rollenmodell zugrunde liegt, werde eine generische Rolle namens „Webseiten-Administrator“ erzeugt und die Zugriffsrechte systemunabhängig definiert. Bei der Einteilung eines Benutzers in diese Rolle können nun diejenigen CMS ausgewählt werden, auf die der Benutzer zugreifen können solle. Dazu ist es nicht nötig, konkret zu wissen, welche Zugriffsrechte vorher für die Rolle definiert wurden.

- **Anforderung $A_{1.3}$: Statischer und dynamischer wechselseitiger Ausschluss.** Einer der Vorteile einer rollenbasierten Zugriffskontrollarchitektur ist, dass sich wechselseitiger Ausschluss von Kompetenzen sehr intuitiv darstellen lässt. Bei diesem als *separation of duty* (SoD) bekannten Prinzip unterscheidet man zwischen statischem und dynamischem Ausschluss, wie bereits in Kapitel 2.2.3 beschrieben wurde. Ebenfalls geht hieraus hervor, dass statisches SoD eine Beschränkung (engl. *constraint*) bei der Benutzer/Rolle-Relation darstellt und wechselseitigen Ausschluss für Rollen definiert, die nicht zusammen an einen einzelnen Benutzer vergeben werden dürfen. Dynamisches SoD dagegen stellt eine Beschränkung der Sitzung/Rolle-Relation dar und definiert diesen Ausschluss für Rollen, die innerhalb desselben Sitzungskontextes nicht gemeinsam aktiviert werden dürfen. Das BRBAC-Modell in dieser Arbeit soll eine Implementierung von dynamischem sowie statischem SoD ermöglichen, ohne dabei den Sitzungskontext als explizites Modellelement zu benötigen, weil dieser implizit bereits vorhanden ist. Ein Beispiel für dynamisches SoD ist dem Bankensektor entliehen, in welchem durch gesetzliche Rahmenbedingungen sehr hohe Anforderungen an die Zugriffskontrolle vorherrschen. In der Geschäftsfunktion eines Bankangestellten sei es beispielsweise möglich, Kredite zu bewilligen und Auszahlungen auszuführen. Zusätzlich solle aber die Restriktion gelten, dass dies beides für einen einzelnen Kredit nicht vom selben Angestellten ausgeführt werden darf, weil es dadurch möglich wäre, dass ein und dieselbe Person einen Kredit gewährt und auszahlt. Ohne dynamisches SoD müssten dafür nun zunächst zwei Rollen definiert werden, die sich durch statisches SoD gegenseitig ausschließen. Dann wäre es allerdings auch nicht mehr möglich, dass ein Bankangestellter sowohl Kredite bewilligen, als auch auszahlen darf. Will man den geschilderten Sachverhalt ohne dynamisches SoD modellieren, ist das nur durch eine explizite Trennung des Kontextes möglich.

Rollenmanagementlösungen, wie sie in den Kapiteln 3.2 und 3.3 vorgestellt wurden, verfolgen auch das Ziel, von den Endsystemen zu abstrahieren, um so die Komplexität zu verringern, die aus der Verwendung von unterschiedlichen Technologien entsteht. Um dies zu erreichen, wird eine zentrale Benutzeroberfläche angeboten, um die Kommunikation zu diesen Systemen durch die Rollenmanagementlösung zu automatisieren. Mit dieser zentralen Forderung abstrahiert man allerdings vom Kontext, was die Verwendung von dynamischem SoD ausschließt. Die technischen Implementierungen von rollenbasierter Zugriffskontrolle basieren auf den NIST-RBAC-Modellen „hierarchisches RBAC“ (siehe Kapitel 2.2.2), bzw. „hierarchischem RBAC mit *constraints*“ (vgl. Kapitel 2.2.3), welche in Übereinstimmung mit dem RBAC-Kernmodell (engl. *core RBAC*, Kapitel 2.2.1) ebenfalls über ein Modellelement für eine Sitzung (engl. *session*) verfügen. Da in technischen Implementierungen aufgrund ihrer systemübergreifenden Architektur vom Sitzungskontext abstrahiert wird, wird dieser heutzutage praktisch

nicht beachtet, wie aus dem ERBAC-Modell und seiner Erweiterung hervorgeht (vgl. Kapitel 3.1.2 und Kapitel 3.1.3). Es gibt hier also eine Diskrepanz zwischen Modellen aus der Wissenschaft und praktischen Umsetzungen in der Industrie. Diese Arbeit schlägt eine Möglichkeit vor, dynamisches SoD ohne einen expliziten Sitzungskontext zu modellieren.

Ziel Z_2: Policy-bezogene Verringerung der Komplexität von Rollen

- **Anforderung $A_{2.1}$: Vererbung von Rechten.** Für die hierarchischen Strukturen, in denen größere Einrichtungen heutzutage organisiert sind, ist es wichtig, dass sie in einem Rollenmodell geeignet abgebildet werden können. Dies ist heute sowohl in Modellen, als auch in kommerziellen Produkten in angemessener Weise vorhanden, wie die Kapitel 2 und 3 belegen. In BRBAC soll die Vererbung von Rechten modelliert werden, was über die Hierarchiebildung hinausgeht, die sich in RBAC und ERBAC widerspiegelt. Durch den hier verfolgten Ansatz werden auch feinere Steuerungsmechanismen für die Vererbung von Rechten ermöglicht. Im Folgenden Absatz wird diese Anforderung motiviert.

  Der Mechanismus der Vererbung geht über die bloße Hierarchiebildung hinaus und ermöglicht eine weitaus effizientere Verwendung von Rollen, sowohl auf geschäftlicher, als auch auf Systemebene. Die Vererbung von Rollen ist bereits in Modellen und technischen Implementierungen vorhanden, jedoch vermischen sich geschäftliche und technische Spezifika durch die fehlende Unterscheidung auf Rollenebene. Eine explizite Trennung, wie sie hier modelliert wird, ermöglicht feinere Steuerungsmechanismen bei der Vererbung. Es lässt sich somit festhalten, dass die Vererbung in Modellen zwar formal spezifiziert und korrekt ist, jedoch den Anforderungen von Unternehmen nach eingängigen Modellen nicht in ausreichendem Maße gerecht wird. In identitätsbasierten Implementierungen existieren Steuerungsmechanismen bei der Vererbung von Rechten, mit denen etwa die Vererbung für spezielle Objekte im Hierarchiebaum umgangen werden kann (engl. *block policy inheritance*), so dass Policys nur für dieses Objekt, nicht aber für die hierarchisch darunterliegenden Objekte angewandt werden [Mic05b]. Ein zweites Beispiel eines feingranularen Steuerungsmechanismus ist das explizite Verbot, dass gewisse Attribute umgangen oder mit anderen Attributwerten belegt werden dürfen (engl. *no override*) [Mic05b]. In einem anschaulichen Beispiel könnte man in einer übergeordneten Rolle fordern, dass gewisse Zugriffsrechte, repräsentiert durch Attribut/Wert-Paare, in hierarchisch darunterliegenden Rollen durch einen abweichenden Werte für dasselbe Attribut nicht überschrieben werden dürfen. Dies wird in der Praxis dadurch verwendet, um gewisse Standards vorzugeben, die für die gesamte Hierarchie gültig sind. Durch den Mechanismus *block policy inheritance* ist es beispielsweise möglich, dass gewisse Rechte in Rollen nicht weitervererbt werden, um zu ermöglichen, dass Rollen über individuelle Rechte verfügen, die nicht Teil der Vererbungshierarchie werden. Der Ansatz des hier vorgestellten Rollenmodells unterscheidet wie bereits erwähnt zwischen der Geschäfts- und Systemebene, was im Hinblick auf die Vererbung einen gezielten Eingriff bei der Rechtevererbung ermöglicht und das Modell insgesamt dynamischer macht und feingranulare Einstellungen im Wirkbetrieb erlaubt. Ein Aspekt, der in bisherigen Modellen bei der Vererbung von Rechten zum Tragen kommt, ist die Steuerung von sich gegenseitig ausschließenden Rechten, die einem Benutzer zuteil werden. Die Modelle und Implementierungen sehen dafür *separation of duty* (SoD) als Mechanismus vor, auf sich gegenseitig ausschließende Rechte einzuwirken. Da dieser Mechanismus nicht unmittelbar mit der Vererbung zu tun hat, wie er in BRBAC verwendet wird, damit aber in Verbindung steht, sei an dieser Stelle darauf hingewiesen, dass SoD in diesem Modell auch Beachtung findet.

- **Anforderung $A_{2.2}$: Unterstützung von *wildcard*-Attributwerten.** Wie technische Umsetzungen von RBAC zeigen, entstehen im Laufe des Lebenszyklus eines Rollenmo-

dells eine Vielzahl von Rollen, was unter anderem daran liegt, dass Rollen immer eine Zusammenfassung von identischen Rechten darstellen und sich in der Praxis aber die Zugriffsrechte der Benutzer unterscheiden. Diese Hypothese stellt die Verringerung von Komplexität, was eines der der Ziele von RBAC ist, als ein Prinzip heraus, das in der praktischen Umsetzung zu mehr Komplexität führt. Dies wirft die Frage auf, ob es Möglichkeiten gibt, dem RBAC-Paradigma treu zu bleiben und dieses Dilemma dennoch zu lösen. BRBAC unterstützt die Verwendung von Platzhaltern für Attributwerte, wie sich auch in den Erweiterungen zum ERBAC-Modell in Form der „benutzerspezifischen Berechtigungen" erwähnt werden. Die Schwachstelle bei dem dort verfolgten Ansatz ist, dass sich die Werte als *constraint* auf der Relation Benutzer/Rolle ausdrücken. Eine Änderung der benutzerspezifischen Berechtigungen zieht somit eine Änderung der Rollenmitgliedschaften nach sich. Im BRBAC-Modell wird diese Schwachstelle umgangen. Nachdem die Anforderung nun spezifiziert und zu anderen Modellen in Bezug gesetzt wurde, wird sie im Folgenden nun genauer betrachtet.

Zunächst wird die Hypothese etwas genauer beleuchtet: Wenn sich die Rechte in einer Rolle lediglich in Nuancen unterscheiden, führt dies im Entwurf zu einer zusätzlichen Rolle, was daran liegt, dass die Zugriffsrechte einer Rolle allgemein als Attribut/Wert-Paare modelliert werden und sich die festgelegten Rechte innerhalb einer Rolle dadurch nicht unterscheiden können. Das Attribut steht hierbei für den eigentlichen Zugriff und der Wert für den Umfang des Zugriffsrechts. Daher müssen zwei Attribut/Wert-Paare, mit unterschiedlichen Wertbelegungen als zwei separate Zugriffsrechte angesehen werden. Wie [Ke02] belegt, unterscheiden sich Rollen oftmals nicht in den Zugriffen selbst, sondern nur im Umfang dieser Zugriffe, was in der hier gewählten Abstraktion von Zugriffsrechten als Attribut/Wert-Paare als unterschiedliche Werte für dieselben Attribute steht. Existieren nun zwei Benutzer, die sich nur in der Belegung dieser Werte unterscheiden, kann dies nicht mehr in einer einzelnen Rolle abgebildet werden. Würde man die beiden Zugriffsrechte in Form von Policys definieren und gemeinsam an eine einzelne Rolle vergeben, hätte sie für einen Zugriff zwei unterschiedliche Zugriffsrechte. Bei einer Autorisierungsanfrage dieser Rolle würde nun entweder stets eines der beiden Recht angewandt, oder die Anfrage könnte nicht bedient werden, weil sie mehrere Ergebnisse zurückliefert. Wollte man den Sachverhalt zweier unterschiedlicher Zugriffsrechte für ein und dasselbe Zugriffsziel formalisieren, müssten demnach zwei eigene Rollen definiert werden. In [Ke02] wird in Form der „benutzerspezifischen Berechtigungen" eine Erweiterung dazu vorgeschlagen, die in dem hier präsentierten Modell aufgegriffen wird. Dabei werden die Zugriffsrechte in Form von Attributen parametrisiert. Diese Erweiterung ermöglicht die Verwendung sogenannter Platzhalter (engl. *wildcards*) für die Werte der Zugriffsberechtigungen in Attribut/Wert-Paaren. Im Gegensatz zu [Ke02] bietet die hier vorgestellte Lösung aber eine zusätzliche Vereinfachung, weil sie bei einer Änderungen des Wertes keine Änderung der Rollen oder deren Aktualisierung nach sich zieht. Die Forderung nach *wildcard*-Attributen kommt somit dem Ziel Z_2 dieses Rollenmodells nach, eine in der Implementierung möglichst effiziente Verwendung von Rolle zu ermöglichen und logisch zusammenhängende Aufgaben, die sich lediglich im Berechtigungsumfang unterscheiden, nicht explizit trennen zu müssen.

- **Anforderung $A_{2.3}$: Automatisierte Rollenmitgliedschaften.** In Anlehnung an die „Joker-Berechtigungen" aus Kapitel 3.1.3 werden in dem hier vorgestellten Rollenmodell BRBAC dynamische Rollenmitgliedschaften eingeführt. Im Gegensatz zu den Joker-Berechtigungen wird mit den automatisierten Rollenmitgliedschaften keine Beschränkung auf den Relationen Benutzer/Rolle und Rolle/Joker-Berechtigung bezeichnet, sondern lediglich auf der Relation Benutzer/Rolle. Auch wird bei Joker-Berechtigungen vorausgesetzt, dass die Syntax des Joker-Attributs im Benutzerobjekt der Namenskonvention entspricht und mit der Syntax in den zugrundeliegenden technischen Systemen übereinstimmt. Dies stellt eine Vermischung von geschäftlichen und technischen In-

formationen dar, was in diesem Modell ja explizit vermieden werden soll, indem beide Sichtweisen differenziert werden. Zudem führt die Forderung nach einer konformen Syntax auf Geschäfts- und Systemebene zu einer Komplexität beim Implementieren eines Rollenmodells, da nicht davon ausgegangen werden kann, dass gewachsene Strukturen dieser Namenskonvention entsprechen. Diese Konformität müsste demnach erst hergestellt werden, was eine Änderung der vorhandenen Strukturen nötig macht. Das Ziel dieser Anforderung ist aber gerade, die Komplexität des resultierenden Rollenmodells zu verringern.

4.2 Trennung von Geschäfts- und Systemsicht

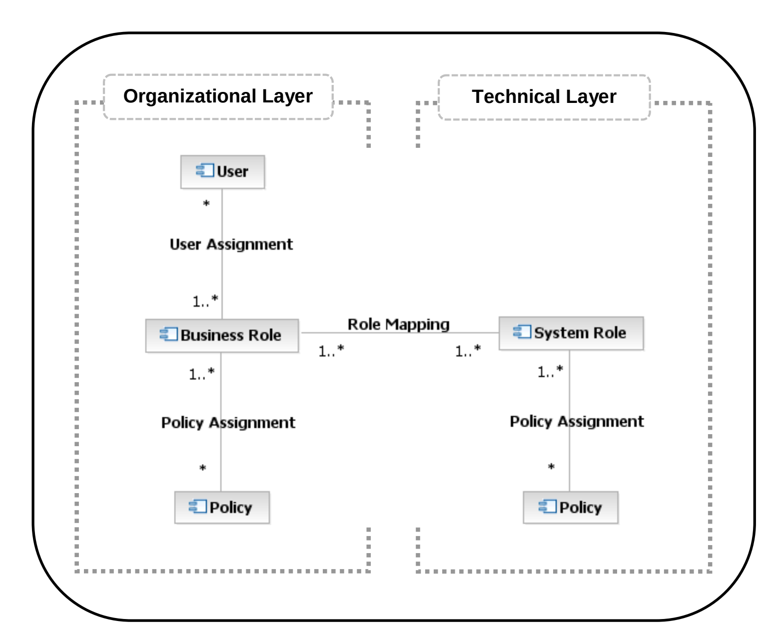
4.2.1 Modellierung von Geschäfts- und Systemrollen

Wie in den Anforderung $A_{1,1}$ zur Modellerweiterung bereits begründet wurde, besteht Nachbesserungsbedarf bei der Definition dessen, was unter einer Rolle verstanden werden kann. Im Kontext heterogener Systeme mit dem Ziel, die Zugriffskontrolle zu homogenisieren, muss zwischen einem geschäftsorientierten und einem technisch geprägten Rollenverständnis unterschieden werden. Diese Unterscheidung hat zur Folge, dass Zugriffsrechte nunmehr danach unterteilt werden, ob sie für geschäftliche Rechte im Sinne von Aufgaben oder Jobprofilen, oder für technische Rechte im Sinne von Zugriffsrechten auf Endsysteme stehen. Dies resultiert in der Spezifikation von Geschäfts- oder System-Policys, die mit dem entsprechenden Rollentyp verknüpft sind. Im Folgenden wird nun zunächst auf diese beiden Begriffe eingegangen und anschließend auf die Verknüpfung zwischen ihnen. Es werden auch Informationen im Hinblick auf mögliche Instanziierungen des Modells gegeben. Information 39 stellt dabei den hierfür benötigten Ausschnitt des Gesamtmodells BRBAC dar.

Unter einer Geschäftsrolle definiert diese Arbeit ein Modellelement, welches eine Geschäftsfunktion in einem Unternehmen repräsentiert. Dazu gehören sämtliche Eigenschaften, die nötig sind, um die geschäftliche Funktion zu erfassen. Beispiele hierfür sind neben dem Titel der Rolle Angaben zu deren Arbeitsplatz, wie etwa die Abteilungszugehörigkeit. Eine Geschäftsrolle kann einer oder mehreren Abteilungen zugeordnet sein, was zur Folge hat, dass Geschäftsfunktionen, die in mehreren Abteilungen gleichermaßen vorkommen, in einer einzigen Geschäftsrolle dargestellt werden können. Dies führt zu kompakten Rollenmodellen auf Geschäftsebene. Auch weist das Modellelement „Geschäftsrolle" eine Hierarchie auf, etwa für einen Verweis auf die Geschäftsrolle des Vorgesetzten, was im Bezug auf Geschäftsprozesse durchaus relevant ist. Damit einhergehend wird eine explizite Rechtevererbung modelliert, auf die im Unterkapitel 4.3.2 „Modellierung der Rechtevererbung" eingegangen wird. Die drei wesentlichen Eigenschaften einer Geschäftsrolle sind allerdings die Benutzermenge, die Menge der Geschäfts-Policys und die Systemrollenmenge. In der Benutzermenge sind alle Benutzerkonten aufgeführt, die die Geschäftsfunktion ausüben. Die Geschäfts-Policys definieren die geschäftlichen Rechte dieser Rolle und in der Menge der Systemrollen stehen alle technischen Zugriffsrechte, die die Geschäftsrolle zur Erbringung dieser Funktion benötigt. Geschäfts-Policys stellen demnach Berechtigungen dar, die nicht auf technischen Endsystemen angewandt werden, sondern die Befugnisse innerhalb von Geschäftsprozessen definieren.

Eine Systemrolle im Gegensatz dazu verkörpert technische Zugriffsrechte in einem oder mehreren Endsystemen. Ihr kann hierbei durch eine geeignete Namensgebung semantische Bedeutung verliehen werden, um das zur Verwaltung nötige technische Verständnis zu verringern. Durch die Kapselung von Geschäftsaufgaben in einem eigenen Rollentyp subsumiert die Systemrolle rein technisches Wissen, was in einem intuitiveren Rollenmodell resultiert. Die beiden wesentlichen Eigenschaften von Systemrollen sind die Geschäftsrollenmenge und die Menge an technischen Zugriffsrechten in den angeschlossenen Endsystemen. Durch die hier eingeführte explizite Trennung von geschäftlichen und technischen Funktionen wird es möglich, eine Rechtevererbung zu unterstützen, auf die in Kapitel 4.3.2 vertieft eingegangen wird. Durch die Einführung eines abstrakt gehaltenen Systemrollenbegriffs, aus dem nicht explizit die darin

enthaltenen Systemberechtigungen hervorgehen, wird das technische Wissen, welches zur Verwaltung einer Rollenmanagementlösung benötigt wird nochmals gesenkt, was die Akzeptanz des Modells in Unternehmen potentiell erhöht.



Information 39: Development of the BRBAC Role Model – Business and System Roles

Die Verknüpfung zwischen den beiden Rollenbegriffen ist so definiert, dass eine Geschäftsrolle eine beliebige Menge an Systemrollen enthalten kann und eine Systemrolle zu einer beliebigen Anzahl von Geschäftsrollen gehört. Im praktischen Einsatz sollten beide Rollen allerdings mindestens eine Referenz auf eine jeweils andere Rolle besitzen. Insbesondere an dieser Stelle kommt die Komplexität von RBAC im Unternehmenseinsatz zum Tragen, weil es hier eine sehr große Zahl an Systemrollen geben kann. Eine manuelle Pflege dieser Relation verursacht einen großen Aufwand, wie schon Information 1 aufgezeigt hat. Die Anwendung von Algorithmen zum Entdecken von Rollen (engl. *role mining*, Kapitel 3.1.1) sowie die Unterstützung von administrativen Prozessen für den Wirkbetrieb (engl. *role management*, *role maintenance*, Kapitel 3.1.4) greifen insbesondere an dieser Stelle an. Speziell das *role mining* als vielversprechender Ansatz für das analytische Entdecken von Zugriffsmustern wird durch die hier eingeführten Rollenbegriffe und der Geschäftsrolle/Systemrolle-Relation unterstützt. In Anlehnung an Information 39 werden folgende Elemente in BRBAC definiert:

- Benutzerkonten (engl. *user*): $U_i, \quad i \in N$;
- Berechtigungen (engl. *policy*): $P_i, \quad i \in N$;
- Rollen: $R \quad \supset BR \cup SR$;
- Geschäftsrollen (engl. *business role*): $BR_i, \quad i \in N$;
- Systemrollen (engl. *system role*): $SR_i, \quad i \in N$;
- Benutzer/Geschäftsrolle-Relation (engl. *user assignment*): $\sigma_{U,BR} \subseteq U \times BR$;
- Geschäftsrolle/Systemrolle-Relation (engl. *role mapping*): $\sigma_{BR,SR} \subseteq BR \times SR$;
- Systemrolle/Berechtigung-Relation (engl. *policy assignment*): $\sigma_{SR,P} \subseteq SR \times P$;

4.2.2 Modellierung von generischen Rollen

Wie in den Anforderungen bereits angeklungen ist, versuchen generische Rollen in BRBAC einerseits ähnliche Rollen zusammenzufassen, um die Anzahl an Rollen zu verringern sowie den technischen Wissensstand zu senken, der notwendig ist, das implementierte Rollenmodell zu verwalten. Generische Rollen stellen eine Verallgemeinerung dar, weil hier von technischen,

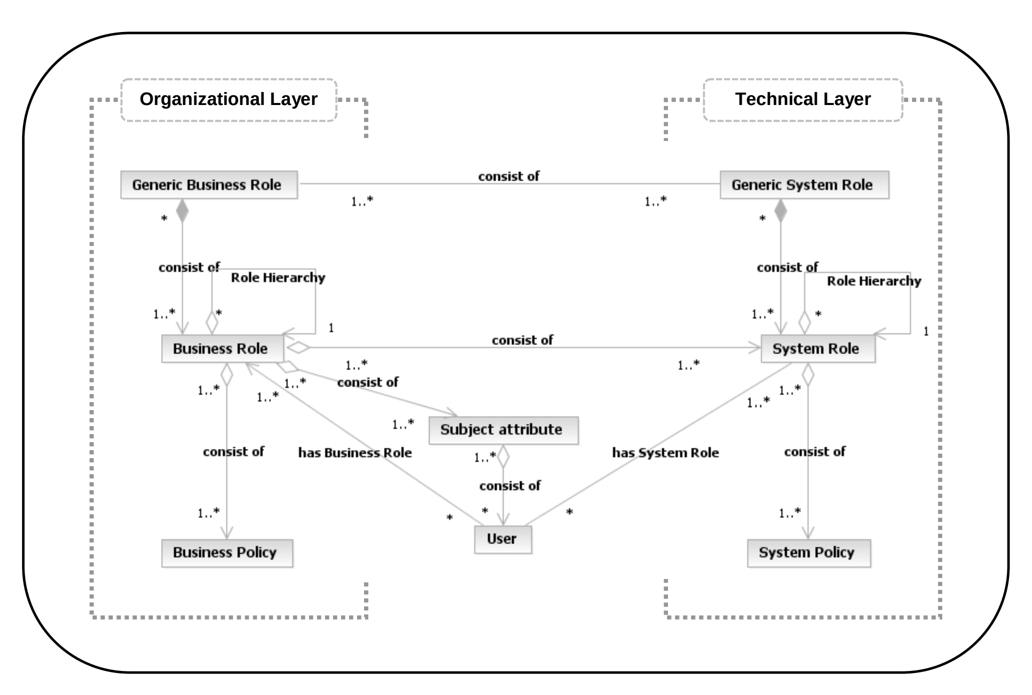
oder konkreten Informationen abstrahiert wird. Durch die zentrale Forderung nach Trennung von Geschäfts- und Systemrollen muss der Begriff der generischen Rollen für diese beiden Rollenbegriffe differenziert betrachtet werden. Im Folgenden wird daher zunächst auf die Generik bei Geschäftsrollen und im Anschluss daran bei Systemrollen eingegangen. Dabei wird im Folgenden das Zielobjekt des Zugriffs einer Rolle als Anwendungsfall bezeichnet.

Da Geschäftsrollen, wie sie bisher verwendet wurden, von sich aus bereits von technischen Details abstrahieren, stellen sich generische Rollen auf Geschäftsebene anders dar, als bei Systemrollen. Generische Rollen verringern in erster Linie die Anzahl an konkreten Rollen dadurch, dass die in ihnen enthaltenen Policys unabhängig vom Anwendungsfall definiert und durch zusätzliche Parameter erst zu einem späteren Zeitpunkt auf diesen festgelegt werden. Überträgt man dieses Prinzip auf Geschäftsrollen, so kann man sagen, dass in ihnen geschäftliche Vorgaben in Form von Geschäfts-Policys die Rechte der Rolle definieren und erst durch die Verknüpfung mit der Organisationsstruktur konkretisiert werden. Hierbei zeigt sich die Nähe von Geschäftsrollen zur Geschäftsprozessmodellierung. Ein Beispiel soll dies verdeutlichen: Man betrachte eine Geschäftsrolle namens „Verwaltungsangestellter", in der durch Policys festgelegt ist, welche organisatorischen Aufgaben ein Verwaltungsangestellter ausübt. Dies kann beispielsweise umfassen, dass er befugt ist, Rechnungen zu unterschreiben und die Überweisung für diese Rechnung bei der Buchhaltung in Auftrag zu geben. Die Festlegung, für welche Rechnungen er autorisiert ist, kann nun entweder in Form von Geschäfts-Policys erfolgen, oder aber auch implizit durch seine Abteilungszugehörigkeit, die in der Geschäftsrolle ja angegeben wird. In diesem Fall verfügt der Verwaltungsangestellte durch seine Abteilungszugehörigkeit über die erforderliche Berechtigung zur Auszahlungsanforderung. Generische Geschäftsrollen greifen genau an diesem Punkt an. Dabei wird die vorher allgemein als Anwendungsfall bezeichnete Abteilungszugehörigkeit aus der Rolle entfernt, wodurch sie allgemeinen Charakter erhält und somit zu einer generischen Rolle wird. Sie besteht aus einer Menge nicht-generischer Geschäftsrollen, die ihrerseits Berechtigungen für konkrete Anwendungsfälle definieren. Dadurch wird deutlich, dass eine generische Geschäftsrolle unterschiedliche Berechtigungsgrade für mehrere Anwendungsfälle repräsentiert. Die Festlegung auf einen konkreten Anwendungsfall geschieht erst bei der Einteilung eines Benutzers in die generische Rolle. Dabei werden aus der Liste zur Verfügung stehender Anwendungsfälle diejenigen ausgewählt, für die der Benutzer Zugriff erhalten soll. Aus dieser Auswahl wird dann die Schnittmenge gebildet aus den Geschäftsrollen, die in der generischen Rolle enthalten sind und den spezifizierten Anwendungsfällen. Das Benutzerkonto wird anschließend in die resultierende Ergebnismenge von Geschäftsrollen eingeteilt. Durch den Algorithmus zur Einteilung in eine generische Rolle und die Festlegung des Anwendungsfalls wird somit sichergestellt, dass ein Benutzerkonto nur über nicht-generische Rollen verfügen kann und generische Rollen in dieser Auslegung als virtuelles Rollenaggregat unterschiedlicher Anwendungsfälle im Rollenmodell enthalten sind. Im Bezug auf die technische Implementierung dieses Rollenmodells muss erwähnt werden, dass der Anwendungsfall einer generischen Geschäftsrolle erst zum Zeitpunkt der Einteilung eines Benutzerkontos angegeben wird und dadurch auf nicht-generische Geschäftsrollen festgelegt wird.

Für Systemrollen stellen sich generische Rollen in anderer Weise dar. Der erwähnte Anwendungsfall steht hier für Endsysteme, die gleiche Zugriffsberechtigungen aufweisen. Gemäß der Definition einer Systemrolle subsumiert sie technische Zugriffsrechte in Endsystemen. Diese können in Form von Policys festgelegt werden. Dabei kann es vorkommen, dass ein und dieselbe Policy für unterschiedliche Systeme definiert wird. Das passiert genau dann, wenn in einem Unternehmen dasselbe technische System mehrfach verwendet wird. Dies führt zu einer Duplizierung von Endsystems mit der gleichen Berechtigungsstruktur, die allerdings durch ihren eindeutigen Bezeichner im Unternehmen nicht als technologisch identisch zu einem weiteren System erkannt werden können. Gerade in größeren Unternehmen mit mehreren unabhängigen Abteilungen ist dies eine durchaus realistische Annahme.

Durch die Verwendung generischer Rollen ist es nun möglich, Systemrollen mit Policys zu definieren, die für alle Instanzen derselben Technologie in gleicher Weise gelten. Außer in der

Verbindung zum konkreten Endsystem unterscheiden sich diese Systemrollen somit nicht und werden in einer einzigen generischen Rolle zusammengefasst. Wird diese generische Systemrolle nun einer Geschäftsrolle zugewiesen, muss die Menge an Endsystemen als zusätzlicher Parameter angegeben werden. Auch hier wird über die Schnittmengenbildung diejenige Teilmenge nicht-generischer Systemrollen errechnet, die in den angegebenen Endsystemen definiert sind und diese dann der Geschäftsrolle direkt zugewiesen.



Information 40: Development of the BRBAC Role Model – Generic Roles

Information 40 illustriert die Modellierung von generischen Rollen. Man erkennt hier, dass die generischen Rollentypen aus einer Menge von nicht-generischen Rollen bestehen. Dies wird auch durch die Teilmengenbeziehung von BR_g und SR_g deutlich. Da eine generische Rolle aus mindestens einer nicht-generischen Rolle bestehen, im Gegenzug dazu eine nicht-generische Rolle aber nicht unbedingt in einer generischen Rolle enthalten sein muss, sind die Mengen BR_g und SR_g echte Teilmengen von BR und SR.

- Generische Geschäftsrollen (engl. *generic business role*): $BR_g \subset BR$;
- Generische Systemrollen (engl. *generic system role*): $SR_g \subset SR$;
- Systemrolle/generische Systemrolle-Relation (engl. *system role assignment*): $\sigma_{SRg,SR} \subseteq SR_g \times SR$;
- Geschäftsrolle/generische Geschäftsrolle-Relation (engl. *business role assignment*): $\sigma_{BRg,BR} \subseteq BR_g \times BR$;
- Geschäftsrolle/Systemrolle-Relation (generisch) (engl. *generic role mapping*): $\sigma_{BRg,SRg} \subseteq BR_g \times SR_g$;

Durch die Kompositionsbeziehung von BR_g mit BR und SR_g mit SR sowie die Relation $\sigma_{BRg,BR}$ ist eine transitive Assoziation von generischen Geschäftsrollen und generischen Systemrollen gegeben, die im praktischen Einsatz allerdings nicht von Belang ist und nur der mathematischen Korrektheit wegen aufgeführt ist. Im Sinne einer Ontologie stellt diese Assoziation eine Verknüpfung zwischen den kontext-unabhängigen Rollenbegriffen zwischen der Geschäfts- und der Systemdomäne dar.

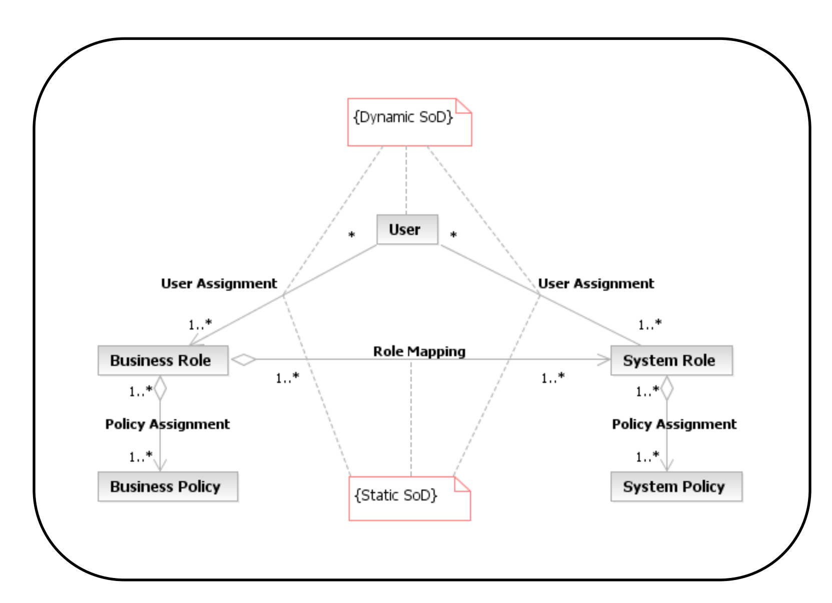
4.2.3 Modellierung von wechselseitigem Ausschluss bei Rollen

Als wechselseitigen Ausschluss (engl. *separation of duty*, SoD) wird ein Prinzip bezeichnet, bei dem definiert werden kann, dass sich Tätigkeitsfelder gegenseitig ausschließen und nicht gleichzeitig an dieselbe Person vergeben werden dürfen. In rollenbasierten Umgebungen sind diese Tätigkeitsfelder als Rollen zu interpretieren, da in ihnen die zur Ausübung der Tätigkeit nötigen Rechte gekapselt sind. Um dies präziser zu formulieren, drücken sich diese Rechte durch Policys aus, die in einer Rolle zusammengefasst sind. Wie schon in Kapitel 2.2.3 gezeigt wurde, ist SoD in rollenbasierten Umgebungen sehr leicht umzusetzen, da hier lediglich Rollen definiert werden müssen, die sich gegenseitig ausschließen. Wie [SC+96] bereits feststellt, ist der wechselseitige Ausschluss die am meisten erwähnte Einschränkung (engl. *constraint*) im Bereich RBAC. SoD bezeichnet dabei ein *constraint* bei der Benutzer/Rolle-Relation. Durch die Auftrennung des Rollenbegriffs gemäß der Anforderung $A_{1,1}$ werden in BRBAC explizit mehrere Relationen im Bezug auf die Rollentypen definiert, was unterschiedliche *constraints* zur Modellierung von SoD möglich macht. Darüber hinaus wird sowohl statisches, als auch dynamisches SoD modelliert, was sich durch *constraints* auf den Relationen Benutzer/Geschäftsrolle ($\sigma_{U,BR}$), Geschäftsrolle/Systemrolle ($\sigma_{BR,SR}$) sowie der transitiven Beziehung Systemrolle/Benutzer ausdrückt.

Wie [FS+01] definiert, unterscheidet man dynamische und statische Aspekte bei SoD. Statisches SoD beschreibt hierbei die Einschränkung, dass gewisse Rollen zur gleichen Zeit nicht an denselben Benutzer vergeben werden können, wohingegen dynamisches SoD zusätzlich dazu den Kontext des Zugriffs betrachtet und der gegenseitige Ausschluss nicht allgemein, sondern nur für bestimmte Kontexte gültig ist. In den gängigen Modellen sind diese beiden Prinzipien bereits formuliert (vgl. hierzu Kapitel 2.2), in den technischen Umsetzungen hingegen fehlt dynamisches SoD bislang gänzlich. Dies liegt daran, dass gerade bei systemübergreifenden Rollen, wie man sie in heterogenen Umgebungen vorfindet, der Kontext verloren geht. Anders als die bisherigen Rollenmodelle für unternehmensweite Zugriffskontrolle ermöglicht BRBAC neben statischen SoD-Aspekten auch dynamisches SoD. Für dynamisches SoD muss der Kontext somit zentral erfasst werden können. Dazu wird dieser im Benutzerobjekt selbst verankert, weil gerade das Benutzerobjekt einen eindeutigen Identifikator eines Benutzers darstellt. Für die Modellierung definiert BRBAC einen Kontext C und zwei Relationen für SoD:

- Kontext: $C \in U_i, \quad i \in \mathbb{N};$
- Statisches SoD-Relation: $\sigma_{sSoD} \subseteq R \times R;$
- Dynamisches SoD-Relation: $\sigma_{dSoD} \subseteq R \times R \times C;$

Wie bereits verbal ausgedrückt wurde, stellt statisches SoD einen *constraint* zwischen zwei Rollen dar und dynamisches SoD zwischen zwei Rollen, die nur in einem bestimmten Kontext gültig sein soll. Dadurch, dass der Kontext nun im Benutzerobjekt selbst verankert ist, kann bei einer Autorisierungsanfrage zur Laufzeit ermittelt werden, ob eine dynamische Einschränkung vorliegt, oder nicht. Diese beiden Konzepte werden in Information 41 grafisch dargestellt.



Information 41: Development of the BRBAC Role Model – Separation of Duties

Der hier eingeführte Ansatz sei durch ein Beispiel verdeutlicht, das dem Bankenwesen entliehen ist, wo klassischerweise sehr starke Einschränkungen vorherrschen. Dazu nehme man an, dass die Rolle „Schalterangestellter" sowohl Kredite bewilligen, als auch die Auszahlung von Krediten anweisen darf. Zusätzlich sei eine Beschränkung definiert, um zu verhindern, dass ein Angestellter in dieser Rolle einen Kredit bewilligt und die Auszahlungsanweisung selbst vornimmt. Damit solle etwa verhindert werden, dass Kredite voreilig vergeben werden, oder ein Angestellter sich selbst einen Kredit bewilligt. Um diesen Sachverhalt formal zu fassen, werden zunächst zwei Policys modelliert, die diese beiden Rechte verkörpern und mit der Rolle „Schalterangestellter" verknüpft. Unter Zuhilfenahme eines statischen *constraint* könnte man nun definieren, dass sich diese beiden Rechte gegenseitig ausschließen, was jedoch in dem gewählten Beispiel dazu führt, dass die Rolle „Schalterangestellter" Kredite entweder bewilligen oder auszahlen darf. Er verfügt dann nur über eines der beiden Rechte, niemals aber über beide. Durch eine dynamische Einschränkung hingegen könnte man definieren, dass sich diese beiden Rechte nicht prinzipiell ausschließen, sondern nur dann, wenn der Kontext für beide übereinstimmt. In diesem Zusammenhang wird nun bei der Autorisierungsprüfung der Kontext der Kreditanweisung überprüft, um festzustellen, ob die Bewilligung des Kredits von derselben Person vorgenommen wurde, die ihn nun auch auszahlen möchte. Der mathematischen Korrektheit halber sei an dieser Stelle erwähnt, dass die Relationen σ_{sSoD} und σ_{dSoD} auf Rollen operieren und nicht auf Policys, wie in diesem Beispiel vorausgesetzt. Da eine Rolle aber eine Menge an Policys subsumiert, lassen sich die Definitionsmengen beider Relationen auf die Policy-Teilmenge der Rollen einschränken.

Zur Umsetzung von dynamischem SoD wird nun ein *constraint* für zwei Rollen, zusammen mit der Bedingung definiert, für die diese Rollen oder die darin enthaltenen Policys nicht übereinstimmen dürfen, wie im erwähnten Beispiel etwa das Benutzerkonto. Dies drückt sich formal in der Relation σ_{dSoD} aus, wobei der Kontext im Benutzerobjekt selbst verankert ist. Jede Aktion, die ein Benutzer ausführt, wird in einem Kontextfeld seines Benutzerkontos hinterlegt, was nebenbei den Effekt hat, dass sämtliche Aktionen dieses Benutzers protokolliert und somit auch noch zu einem späteren Zeitpunkt nachvollzogen werden können. Damit kommt dieses Modell rechtlichen Vorgaben nach, nach denen Benutzeraktionen lückenlos protokolliert werden müssen. An dieser Stelle sei verwiesen auf den *Sarbanes-Oxley Act* (SOX) [Sen02] und das Gesetzt zur Kontrolle und Transparenz im Unternehmensbereich (KonTraG) [Bun98]. Das Ergebnis dieses Ansatzes ist eine erhöhte Sicherheit bei vertraulichen Arbeitsgängen unter Beachtung des jeweiligen Kontexts. Das betrifft sowohl geschäftliche Vorgängen wie etwa Überweisungen/Kreditbewilligung als auch tiefgreifende Eingriffe ins System, wie etwa dem Löschen eines Benutzerkontos.

4.3 Modellierung von Policy-Erweiterungen

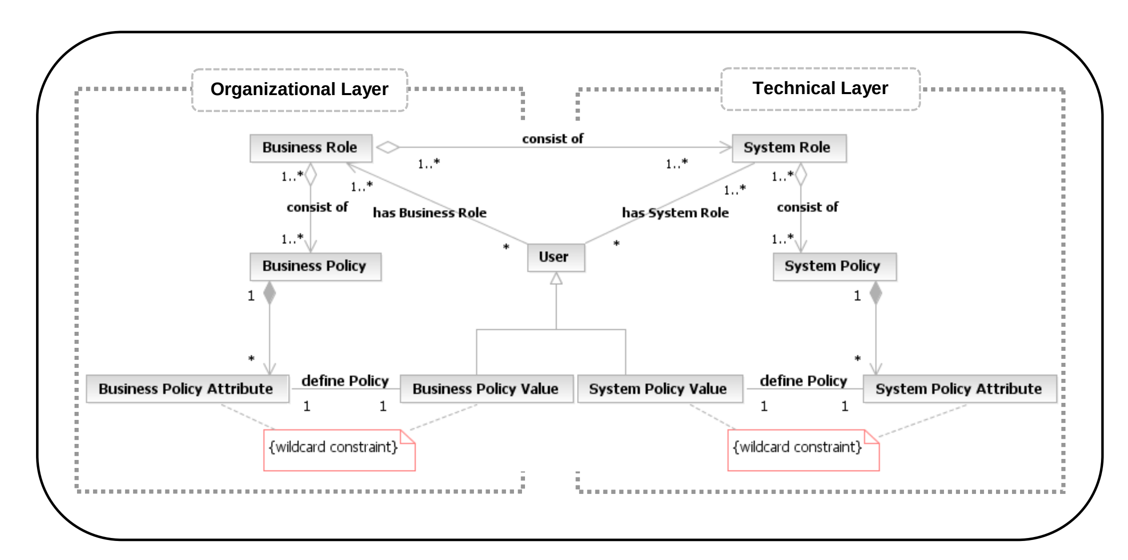
In den vorangegangenen drei Teilkapiteln lag der Fokus auf der Modellierung der Anforderungen für das Ziel Z_1. Dabei standen die Unterteilung von Geschäfts- und Systemrollen im Vordergrund sowie der Zusammenhang zwischen diesen beiden Begriffen. Die eingehende Betrachtung der beiden Rollenbegriffe stellt den Teil der Modellspezifikation von BRBAC dar, wohingegen in den verbleibenden drei Teilkapiteln auf Mechanismen eingegangen wird, die zu einer effizienten Verwendung dieser Rollen ausgehend von Policys führt, was sich an das Ziel Z_2 wendet. Dieses Kapitel befasst sich nun mit der Modellierung der drei Anforderungen $A_{2,1}$, $A_{2,2}$ und $A_{2,3}$, wie im Einführungskapitel vorgestellt wurde; im Speziellen geht es dabei um die Policys, die in den Rollen enthalten sind und die den Grad an Zugriffsrechten spezifizieren.

4.3.1 Modellierung von Policys mit Platzhaltern

[Em08] bezeichnet eine Policy als Inbegriff von Statuten, Richtlinien oder Verhaltensregeln für die Benutzung von Systemen, Netzwerken oder Diensten. In diesem Zusammenhang steht eine Policy für eine Abbildung von einer Menge an Attributen auf einen Wahrheitswert, der darüber Aussagen trifft, ob der Zugriff gewährt wird, oder nicht. In Anlehnung an diese Definition stellt eine Policy hier eine Relation zwischen verschiedenen Attributen und deren Werten her.

- Policy (engl. *policy*): $\sigma_{\mathrm{Pol}} \subseteq \bigcup(ATTRIBUTE \times VALUE)$;

Das Ziel, was in diesem Teilkapitel verfolgt wird, ist eine möglichst effiziente Verwendung von Rollen zu ermöglichen. Es stellt sich dabei die Frage, inwieweit Modellierungsaspekte bei Policys hierfür herangezogen werden können oder sollen. Geschäfts- und System-Rollen bestehen gleichermaßen aus Policys, die im jeweiligen Anwendungsfall konkrete Zugriffsrechte auf atomarer Ebene darstellen. Diese Arbeit will kein explizites Policy-Modell entwickeln, aber durchaus Anforderungen an Policys stellen, die zu einem kompakten Rollenmodell und dadurch zum Erreichen von Ziel Z_2 führen. Dazu werden Policys nur soweit betrachtet, wie in diesem Kontext nötig, da die Policy-Modellierung außerhalb des Fokus dieser Arbeit steht. Verallgemeinernd und in Übereinstimmung mit der Definition der Relation σ_{Pol} wird eine Policy als 2-Tupel definiert, bestehend aus einem Autorisierungsattribut und dessen Wert. Diese Abstraktion ist hinreichend, weil mindestens diese beiden Informationen erfasst werden müssen, um eine Zugriffskontrollaussage treffen zu können. In diesem Kapitel wird nun gemäß der Anforderung $A_{2,1}$ speziell auf den Wert eines Autorisierungsattributs eingegangen und dieser Sachverhalt modelliert. Da sowohl Geschäfts- als auch System-Policys als 2-Tupel gesehen werden können, ist das in diesem Teilkapitel vorgestellte Prinzip der Autorisierungsattribute mit Platzhaltern für beide Rollenbegriffe gleichermaßen gültig. Information 42 stellt Policys als Attribut/Wert-Paare mit Platzhaltern (engl. *wildcards*) dar.



Information 42: Development of the BRBAC Role Model – Wildcard Attributes

Durch das 2-Tupel ist ein Zugriffsrecht formal spezifiziert. Im RBAC-Umfeld werden jedem Benutzer Zugriffsrechte durch seine Rollenmitgliedschaften zugeteilt, die in den Policys definiert sind. Diese Trennung von Benutzern und Rechten stellt das grundlegende Ziel von RBAC dar. Dadurch wird erreicht, dass Rechte nur in konsistenter und nicht in individueller Form vergeben werden. Auch werden dadurch Policy-Vorgaben effektiv umgesetzt. Dies führt allerdings in Situationen, in denen individuelle Attributwerte vergeben werden, unausweichlich dazu, dass hierfür eigene Policys erzeugt werden müssen. Betrachtet man diese Policys genauer, so stellt man fest, dass sie sich oftmals nicht in den enthaltenen Attributen unterscheiden, sondern lediglich in den Attributwerten. Wie in den Erweiterungen zum ERBAC-Modell belegt wird, führt dieser Ansatz genau in solchen Fällen zu einer Vervielfachung von Rollen, wo sich die Zugriffsmuster nicht im Attribut, sondern nur in dessen Wert unterscheiden.

Es stellt sich die Frage, ob sich dieses Problem vereinfachen lässt, denn offenbar sind diese Rollen bis auf den konkreten Attributwert identisch, oder zumindest sehr ähnlich. Das hier vorgestellte Modell BRBAC verknüpft den zum Autorisierungsattribut gehörenden Wert nun nicht mit der Policy selbst, sondern mit dem Benutzerkonto. Wie in der Modellierung von Anforderung $A_{1,3}$ wird auch hier der Kontext, der in diesem Fall durch den Wert des Autorisierungsattributs festgelegt ist, von der Policy getrennt. Erst durch die Verknüpfung von Policy und Benutzer wird somit das spezifizierte Zugriffsrecht klar. Der Vorteil, der durch diesen Ansatz entsteht, liegt klar auf der Hand: Durch die Trennung des Wertes ist es möglich, in einer Policy ein Autorisierungsattribut zu definieren, dessen konkreter Wert von den individuellen Benutzern abhängt. Der Nachteil davon ist, dass die Verwaltung dieses Modells mit großer Sorgfalt erledigt werden muss, da Platzhalter nicht in jedem Szenario gleichermaßen angewandt werden können. Mit diesem Ansatz wird das Ziel Z_2 zur Komplexitätsreduktion auf Policy-Basis erreicht, denn es werden Rollen zusammengefasst, deren Policys sich nur in den Attributwerten unterscheiden.

Abschließend wird auch hier ein Beispiel zur Verdeutlichung gegeben: In einer Bank verfügen Angestellte in der Rolle „Schalterangestellter“ über unterschiedlich hohe Bewilligungsgrenzen für Kredite. Normalerweise führt dies zu zwei eigenständigen Rollen, die jeweils über eine Policy verfügen, die die unterschiedlichen Bewilligungsgrenzen dieser Tätigkeit definieren. Durch den hier vorgestellten Ansatz ist es möglich, lediglich das Recht zur Bewilligung von Krediten für die Rolle „Schalterangestellter“ in einer Policy festzulegen und durch einen *wildcard*-Wert im Autorisierungsattribut den entsprechenden Wert vom Benutzerkonto zu beziehen, was unmittelbar zur Laufzeit geschehen kann. Wie bereits erwähnt wurde, stellt dieser Ansatz Anforderungen an die Sorgfalt, mit der das resultierende Rollenmodell gepflegt wird. Die Verwendung von Policys mit Platzhaltern stellt keine prinzipielle Vorgehensweise dar, sondern lediglich eine Möglichkeit, die unter gewissen Umständen als sinnvoll erachtet werden kann.

Diese Umstände gilt es, individuell zu diskutieren. In diesem Beispiel führt der vorgestellte Ansatz zur Entwicklung von lediglich einer Rolle „Schalterangestellter“ mit der Policy „Kreditbewilligung“. Anders als in den Erweiterungen zum ERBAC-Modell aus [Ke02], wo dieser Mechanismus auch verwendet wird, bleiben die konkreten Werte im Benutzerobjekt erhalten. In [Ke02] werden diese Werte zu dem Zeitpunkt, zu dem ein Benutzer in die Rolle eingetragen wird, in die Policy fest eingetragen und bei einer Änderung des Wertes durch einen vorher definierten Algorithmus angepasst. Dadurch, dass der Attributwert selbst im Benutzerkonto enthalten ist, hat eine Änderung der Rollen über deren Lebensdauer keinerlei Auswirkung auf die Policys, außer natürlich, die Policys werden selbst geändert. Auch kann hierdurch für jeden Benutzer individuell eine andere Grenze verwendet werden, die zur Laufzeit bestimmbar ist, was das Rollenmodell an dieser Stelle sehr dynamisch macht. Sämtliche Rollen, die über identische Zugriffsmuster verfügen, sich jedoch nur im Zugriffsumfang unterscheiden, können nun als eine einzelne Rolle dargestellt werden.

4.3.2 Modellierung der Rechtevererbung

Nach einer Betrachtung von Policys als Modellelement zur Kapselung von Zugriffsrechten wird nun auf die Möglichkeit eingegangen, diese Rechte in Hierarchien anzuordnen und – darauf aufbauend – eine explizite Vererbung von Rechten zu modellieren. Der heutige Stand bei der Vererbung von Rechten basiert auf Hierarchien und bedeutet, dass die Policys in Rollen entlang der Hierarchie weitervererbt werden. Dieser Mechanismus wurde bisher im RBAC-Kontext keiner differenzierteren Betrachtung unterzogen. In vielen RBAC-Umsetzungen wird die Rollenmitgliedschaft statisch weitergegeben, es sei denn, die vererbten Rechte würden in Konflikt stehen zu einem SoD-*constraint*, der die Kombination dieser Rechte ausschließt. Wie in Kapitel 6.3 gezeigt wird, gibt es sogar aktuelle technische Umsetzungen, die die Vererbung innerhalb der Hierarchien nur zum Teil umsetzen. Aus IBAC-basierten Umgebungen sind jedoch feinere Steuerungsmechanismen bekannt, die auf RBAC übertragen werden (vgl. [Mic05b]). Zur Tragfähigkeit werden im Folgenden die zwei Prinzipien *block policy inheritance* und *no override* erläutert, die anschließend in BRBAC modelliert werden.

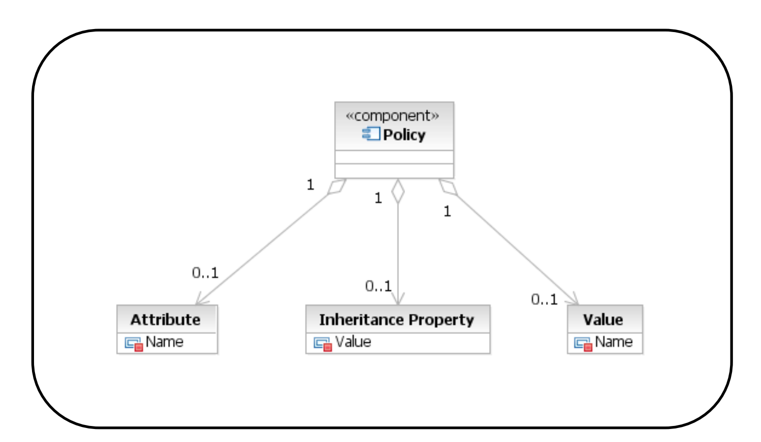
- **Umgehen der Policy-Vererbung.** Die zentrale Trennung von geschäftlichen und technischen Aspekten kommt auch an dieser Stelle zum Tragen, so dass explizit zwischen geschäftsnahen und technischen Policys unterschieden wird, was zu einer klaren Struktur führt. Der Ansatz, der hier verfolgt wird, ist, dass gewisse Policys vom Vererbungsmechanismus explizit ausgenommen sind. Dazu wird auf die Vererbung in sofern eingewirkt, als dass gesteuert werden kann, ob gewisse Rechte vererbt werden, oder nicht. Dazu ist es in diesem Modell möglich, in einer Policy zu definieren, ob das darin spezifizierte Recht weitergegeben wird oder nicht. Dazu erhält eine Policy ein zusätzliches Attribut *block policy inheritance*, welches den geschilderten Sachverhalt spezifiziert. Dieses Attribut ist hierbei direkt in der Policy und nicht in der Rolle selbst verankert, um eine feingranulare Steuerung zu ermöglichen. Da SoD-*constraints* auf der Ebene von Rollen operieren, muss dies bei der Implementierung des Rollenmodells beachtet werden. Durch diesen Mechanismus ist es nunmehr möglich, gewisse Policys explizit von der Vererbung auszuschließen, während die Rollen, die diese Policys enthalten, weitervererbt werden können. Der Vorteil hiervon ist, dass eine Rolle innerhalb der Hierarchie individuelle Rechte zugewiesen bekommen kann, die nur für diese Rolle gelten, nicht aber für die Rollen in tiefer liegenden Hierarchiestufen.

- **Überschreiben bei gleichen Zugriffsrechten verbieten.** Ein zweiter Steuerungsmechanismus wird als *no override* bezeichnet. Durch die Vererbung von Rechten ist es möglich, dass gewisse Rechte mehrfach vergeben werden und sich im Zugriffsumfang unterscheiden. Als Beispiel hierfür sei ein Szenario gegeben, in dem der Zugriff auf ein spezielles Endsystem in zwei Rollen definiert sei. In der in Kapitel 4.3.1 gewählten Abstraktion einer Policy als Attribut/Wert-Paar entspräche dieser Sacherverhalt zwei Policys, die sich nicht im Autorisierungsattribut, sondern im Attributwert unterscheiden.

Erhielte ein Benutzer diese beiden Rollen, verfügte er somit über zwei Policys, die für dasselbe Recht einen unterschiedlichen Rechteumfang definieren. Falls dieser Konflikt erkannt würde, könnte damit in unterschiedlicher Weise umgegangen werden: Durch die Bildung von Vereinigungs- oder Schnittmengen würde dem Benutzer entweder die restriktivere oder die umfangreichere Rechtemenge zuteil. Eine dritte Möglichkeit, mit diesem Konflikt umzugehen ist, einen Verantwortlichen zu informieren und bis dahin beide Rechte zu gestatten oder zu unterbinden. Da die Vergabe von widersprüchlichen Rechten im Allgemeinen nicht beabsichtigt geschieht, führen alle diese geschilderten Ansätze zumindest potentiell zu Missverständnissen. In diesem Modell wird durch das Attribut *no override* eine Möglichkeit gegeben, mit mehrfach definierten Rechten effizient umzugehen. Dazu wird davon ausgegangen, dass in der Hierarchie weiter oben definierte Rechte mehr Gewicht haben und von tiefer liegenden Policys nicht überschrieben werden können. Dies löst die Mehrdeutigkeit bei der Vererbung auf und ermöglicht es darüber hinaus, gewisse Rechte für den gesamten Hierarchiebaum vorzugeben und durchzusetzen. Es dient daher sowohl der feingranularen Steuerung von Rechtevererbungen, als auch zur Verringerung von Komplexität. Durch dieses Attribut kann festgelegt werden, dass gewisse Rechte von hierarchisch tieferliegenden Rollen nicht überschrieben werden können.

Nachdem diese beiden Mechanismen eingeführt wurden, soll nun auf das Zusammenspiel von Policy-Vererbung und SoD-*constraints* eingegangen werden, um aufzuzeigen, was bei technischen Umsetzungen beachtet werden muss. Bisher wurden mit SoD und der Rechtevererbung zwei Techniken vorgestellt, die dem Rollenmodell Dynamik verleihen. Dadurch müssen verschiedene Szenarien betrachtet werden, in denen diese Techniken zusammentreffen, um daraus abzuleiten, ob es hierbei zu Komplikationen kommen kann. Wie bereits beschrieben wurde, operiert SoD auf der Basis von Rollen und die Rechtevererbung auf der Basis von Policys. Es kann demnach passieren, dass eine Rolle aus einer Menge von Policys besteht, die ihrerseits mit den beiden erwähnten Attributen *block policy inheritance* oder *no override* versehen sind. Durch das Attribut *block policy inheritance* wird die Policy explizit von der Vererbung ausgenommen. Bei der Implementierung des Rollenmodells muss demnach darauf geachtet werden, dass diese Policys nicht an die hierarchisch tiefer liegenden Rollen weitergegeben werden. Falls das Attribut *no override* in der Policy einer Rolle vergeben wurde und zusätzlich ein SoD-*constraint* zwischen dieser Rolle und einer hierarchisch tiefer liegenden Rolle definiert wurde, kommt die höherwertige Policy trotz des *no override*-Attributs nicht zur Anwendung, weil die Rolle, die diese Policy enthält, eine Policy-Verletzung verursacht somit wegen des SoD-*constraint* nicht verteilt wird. Aus dieser Betrachtung erkennt man, dass die hier eingeführten Definitionen die Konsistenz wahren und zu keinen Komplikationen führen.

Diese beiden Mechanismen finden sich bislang weder in Modellen, noch in RBAC-Umgebungen wieder, ermöglichen aber eine sehr feine Steuerung der Rechtevererbung. Durch diese Vererbung wird das Ziel Z_2 unterstützt. Sie ist in Information 43 am Beispiel einer Policy illustriert.



Information 43: Development of the BRBAC Role Model – Policy Inheritance

Auch an dieser Stelle sei abschließend ein Beispiel angegeben. Man gehe von einem Unternehmen aus mit drei Hierachieebenen: Der Geschäftsleitung, den einzelnen Abteilungen mit Abteilungsleitern sowie den Bediensteten in den Abteilungen. Die Geschäftsleitung habe bei der Anmeldung an einem System die Startseite des Internet-Browsers auf die externe Firmen-Homepage gesetzt. Da dies eine technische Einstellung mit Bezug zu einem technischen Endsystem darstellt, würde das Beispiel als System-Policy realisiert, die sich entsprechend an alle Hierarchiestufen weitervererbt. Die Abteilungsleiter jedoch besitzen eine interne Kollaborationsplattform, die als Startseite verwendet werden solle. Dazu müsste man eine zweite Policy definieren, die als Wert statt der Firmen-Homepage nun die Kollaborationsplattform hat. Man hätte demnach zwei System-Policys, die unterschiedliche Werte für dasselbe Attribut definieren. Ginge man davon aus, dass die zweite Policy die erste überschreibt, werden sowohl die Abteilungsleiter als auch die Mitarbeiter der Geschäftsleitung korrekt umgeleitet. Allerdings hätte man dann das Problem, dass die Angestellten der Abteilung ebenfalls auf das Portal umgeleitet werden, weil diese in der Hierarchie unter den Abteilungsleitern angeordnet sind. In dem hier vorgestellten Modell kann man nun die entsprechende Policy auf Abteilungsebene durch das Attribut *block policy inheritance* versehen, so dass die Einstellung für das Kollaborationsportal nur für diese Hierarchiestufe angewandt und nicht an die Angestellten weitervererbt wird. Als Beispiel für das Attribut *no override* gehe man davon aus, dass dieses Unternehmen im Sinne einer gemeinsamen Unternehmensidentität (engl. *corporate identity*) erreichen möchte, dass alle Benutzer auf die Startseite der Firma umgeleitet werden, ungeachtet dessen, was in den einzelnen Hierarchiestufen definiert ist. Dazu genügt es in diesem Modell, die Policy auf oberster Ebene mit dem Vererbungsattribut *no override* zu versehen, um diese Einstellung uneingeschränkt zu vererben. Ohne die Verwendung der Vererbungsmechanismen würde dieses Beispiel als drei eigenen Systemrollen mit jeweils einer Policy umgesetzt. Stünden diese Systemrollen in einer hierarchischen Beziehung zueinander, wäre das skizzierte Beispiel bereits nicht mehr modellierbar, oder man arbeitet mit damit, dass sich Berechtigungen gegenseitig überschreiben können. Dies würde der Intuitivität des Rollenmodells sowie seiner klaren Strukturierung jedoch entgegenwirken.

Das Ergebnis dieses Ansatzes ist die Modellierung der Rechtevererbung, die sehr klar strukturiert ist. Zusätzlich dazu kann die Vererbung durch die Attribute gesteuert werden. Die Auswertung der effektiven Rechte eines Benutzerkontos findet hierbei dadurch statt, dass die Vereinigungsmenge aus den einzelnen Policys unter Berücksichtigung von *constraints* und den Vererbungsattributen gebildet wird.

4.3.3 Automatisierte Rollenmitgliedschaften auf der Basis von Policys

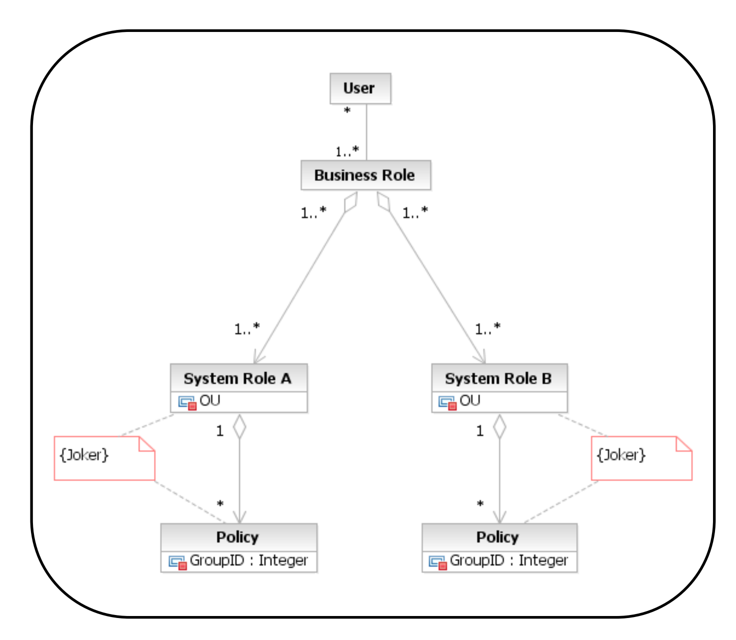
Bisher wurde zunächst die Aufteilung von Geschäfts- und Systemrollen betrachtet und im Anschluss daran auf die Auswirkungen auf Policys eingegangen. In diesem abschließenden Teilkapitel soll nun ein Aspekt betrachtet werden, der insbesondere die Verwaltung des Rollenmo-

dells im Wirkbetrieb unterstützt. Dabei handelt es sich um eine Automatisierung von Policy-Definitionen. In dynamischen Unternehmen, wie man sie heute vorfindet kommt es häufig vor, dass ein Angestellter in unterschiedlichen Dienststellen arbeitet. In diesem Zusammenhang muss es möglich sein, dass er in der Ausübung seiner geschäftlichen Rolle in den unterschiedlichen Dienststellen über dieselben Rechte verfügt. Da die Dienststellen durch die intensive Nutzung des Internet heutzutage zu einem gemeinsamen Netzwerk verbunden sind, ist die Grundlage für diese Betrachtung faktisch schon gegeben. Die unterschiedlichen geschäftsnahen Strukturen wie etwa Geschäftsrollen sind dadurch bereits im gesamten Unternehmen abrufbar. Anders stellt sich diese Situation jedoch dar, wenn man nicht die geschäftliche Ebene betrachtet, sondern die Ebene der Zugriffsrechte. Da in den unterschiedlichen Abteilungen an unterschiedlichen technischen Systemen gearbeitet wird, kann es passieren, dass in diesen Abteilungen individuelle *constraints* bei technischen Berechtigungen vorhanden sind. Die technischen Rechte hängen dann nicht nur von der Rolle selbst ab, die ein Benutzer ausübt, sondern auch von anderen Einschränkungen, wie etwa dem Standort. Diese Arbeit greift dabei den Mechanismus der „Joker-Berechtigungen“ aus dem ERBAC-Modell aus [Ke02] auf. Diese Form der Berechtigung verfügt über *constraints*, die als Bedingung aufgefasst werden können. Ein Benutzer kann in die mit der „Joker-Berechtigung“ verknüpfen Rolle nur eingeteilt werden, wenn die Bedingung erfüllt ist. Der interessante Unterschied zu gewöhnlichen *constraints* besteht darin, dass die Bedingung nicht statisch festgelegt ist, sondern von einem Attributwert im Benutzerkonto abhängt. In [Ke02] wird dies dadurch modelliert, dass die Bedingung von einem Attribut im Benutzerkonto abhängt und der Wert dieses Attributs syntaktisch der Nomenklatur eines speziellen Endsystems entspricht. In Abhängigkeit vom Attributwert wird der Benutzer dann in eine entsprechende Gruppe eingeteilt. Das bedeutet insbesondere, dass diejenigen Mitarbeiter des Unternehmens, die die Benutzerkonten konfigurieren, die Architektur des Endsystems kennen und verstehen müssen, um den Attributwert an die Namensgebung des Endsystems anpassen zu können, wovon in der Regel nicht ausgegangen werden kann. Dadurch vermischen sich Spezifika aus der geschäftlichen und der technischen Ebene. Diese Schwachstelle wird in BRBAC umgangen. Somit kommt diese Anforderung $A_{2,3}$ dem Ziel Z_2 nach. Die zweite Schwachstelle der „Joker-Berechtigungen“ ist, dass die Auswertung der Regeln zu dem Zeitpunkt geschieht, zu dem ein Benutzer in eine Rolle eingeteilt wird. Wenn sich an dem Benutzerattribut Änderungen ergeben, zieht das keine Neuauswertung der Regel nach sich, was potentiell zu Inkonsistenzen führt, wenn diese Änderung nicht manuell auch in den Rollenzuweisungen beachtet werden. In dem hier verfolgten Ansatz soll die Einschränkung allerdings nicht statisch gültig sein, sondern die Möglichkeit bieten, dass sie nur für eine bestimmte Zeit gültig ist und automatisch rückgängig gemacht werden. Daher sollte die Einteilung in diese systemspezifische Rolle nicht bei einer Änderung der Rollenmitgliedschaften geschehen, sondern jedes mal, wenn sich ein Benutzer anmeldet, da an dieser Stelle sein Kontext festlegt wird.

Nachdem der Mechanismus erklärt wurde, soll er nun in BRBAC modelliert werden. Es wurde bereits erwähnt, dass es sich hier um rein technische Spezifika handelt, weswegen hier lediglich von Systemrollen und System-Policys die Rede ist und nicht von Geschäftsrollen und deren Policys. Eine Joker-Policy unterscheidet sich von einer gewöhnlichen Policy dadurch, dass sie mit einer Regel versehen ist, die bei jeder Autorisierungsanfrage überprüft und ausgewertet wird. Jede Systemrolle besteht aus einer Menge an System-Policys, die ihrerseits durch ihre Attribute auf Endsysteme festgelegt sind. Will man eine Rolle definieren, die systemübergreifende technische Rechte verkörpert, so hat man dazu prinzipiell zwei Möglichkeiten: Entweder definiert man mehrere Policys zu einer Systemrolle, die ihrerseits die technischen Rechte auf ein Endsystem festlegt, oder man definiert mehrere Systemrollen, die über die Policys für ein konkretes Endsystem verfügen. Durch den letztgenannten Ansatz wird die Grenze zwischen den Systeme hervorgehoben, weil sie sich in den Systemrollen ausdrückt.

An dieser Stelle muss klar zwischen generischen Rollen aus Kapitel 4.2.2 und Joker-Policys unterschieden werden; beide Prinzipien abstrahieren von systemspezifischen Details, jedoch mit unterschiedlicher Ausprägung. Verfügen die erwähnten Endsysteme über dieselben Rechte, bedient man sich generischer Rollen, um vom konkreten System zu abstrahieren. Der Anwen-

dungsfall wird dann erst durch die in der generischen Rolle aufgeführten Systeme festgelegt. Die Platzhalter bei Policys aus Kapitel 4.3.1 ergänzen diesen Ansatz, weil hierdurch von den konkreten Attributbelegungen abstrahiert wird. Was hiermit allerdings nicht möglich ist, ist das Zusammenfassen von Endsystemen, für die unterschiedliche Rechte definiert sind.



Information 44: Development of the BRBAC Role Model – Automatic Roles

Wenn ein Benutzer an unterschiedlichen Standorten gleichermaßen arbeitet, bedeutet das für die Zugriffsrechte, dass seine Geschäftsrolle über eine große Zahl an Systemrollen verfügt, die ja gerade den Zugriff auf die Endsysteme verkörpern. Dies wird in Information 44 durch „System Role A“ und „System Role B“ ausgedrückt. Die Verwaltung dieser Rolle kann nun dadurch dezentralisiert werden, dass die einzelnen Abteilungen die atomaren Zugriffsrechte auf ihren eigenen Endsystemen selbst definieren und an eigene Systemrollen knüpfen. An dieser Stelle wird klar, wieso generische Rollen für dieses Szenario nicht verwendet werden können: Diese Rollentypen dienen einer Zusammenfassung gleicher Rechte in unterschiedlichen Systemen, hier jedoch handelt es sich um unterschiedliche Rechte in unterschiedlichen Systemen. Es ist eine realistische Annahme, dass Gruppenzugehörigkeiten in technischen Endsystemen in Relation stehen zur Abteilungszugehörigkeit. Daraus kann man schließen, dass sich die Zugehörigkeit zu Gruppen, die in Endsystemen vorhanden sind, automatisch bzw. automatisiert aus der Abteilungszugehörigkeit ableiten lässt. Es besteht demnach eine Relation zwischen Abteilungs- und Systemzugehörigkeit. Dies kann durch eine Regel automatisiert werden. So wie bei Policys durch Platzhalter-Attribute ein Verweis auf individuelle Benutzerattribute realisiert wird, die zur Laufzeit ausgewertet werden, wird bei Policys durch Joker-Werte der Bezug zur Organisation angegeben. Obwohl sich die Prinzipien hier ähneln, ist die Zielsetzung dabei jedoch eine ganz andere: Während bei Platzhalten die Dynamik im Wirkbetrieb unterstützt wird, können durch Joker-Attribute Rollen auf der Basis von Regeln automatisch erzeugt werden. Dies fällt somit in den Bereich der Rollenmodellierung und nicht des Rollenmanagements.

Zum Abschluss soll auch dieser Sacherverhalt durch ein Beispiel verdeutlicht werden. Ein Unternehmen habe mehrere Abteilungen, die technische Systeme eigenständig betreiben. Ein Mitarbeiter in der Rolle „Administrator“ arbeite in allen diesen Abteilungen gleichermaßen. Dazu benötige er die Systemrollen „SR_AbteilungA“, „SR_AbteilungB“ und „SR_AbteilungC“, wobei für die Verwaltung dieser Rollen jede Abteilung selbst zuständig ist. Dabei subsumieren diese drei Systemrollen die Rechte, die in der Ausübung der Rolle „Administrator“ in den drei Abteilungen benötigt werden. Ohne Joker-Berechtigungen verfügt die Rolle „Administrator“ über drei Systemrollen mit den darin definierten Policys. Verwendet man die hier modellierten

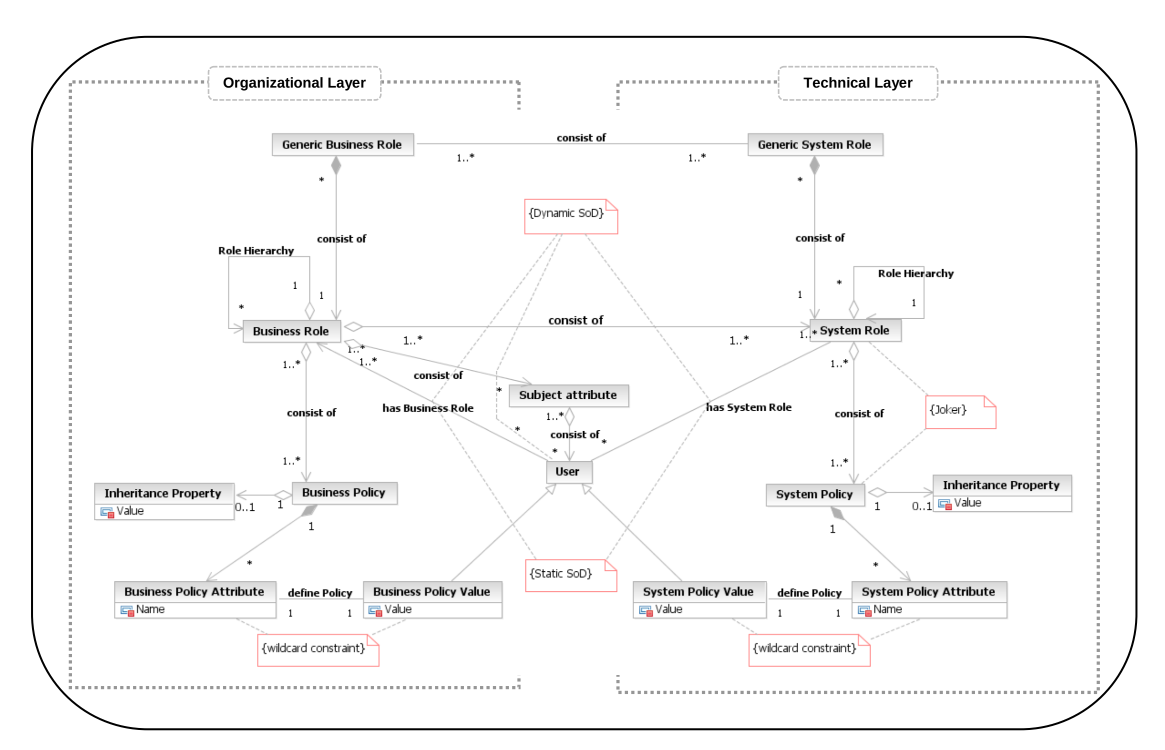
Joker-Policys, werden lediglich die Policys definiert und die Erzeugung von Systemrollen kann unabhängig davon automatisiert geschehen.

4.4 Gesamtmodell

In diesem Kapitel wurde mit BRBAC ein Rollenmodell für den Einsatz in heterogenen Umgebungen entwickelt, wie man sie typischerweise in Unternehmen vorfindet. Es soll abschließend in seiner Gesamtheit dargestellt wird. Hierfür werden die Hauptziele und die daraus abgeleiteten Anforderungen nochmals zusammengefasst und auf Möglichkeiten eingegangen, die sich aus der Kombination der hier präsentierten Modellierungen ergeben.

Das zentrale Ziel des BRBAC-Modells ist die Trennung der geschäftsnahen von den technischen Rollen. Den bisherigen Rollenmodellen fehlt diese klare Trennung, was zu Modellen führt, die in der praktischen Umsetzung schlecht zu handhaben sind. Der Rollenbegriff, der in diesen Modellen verwendet wird, ist zu allgemein gefasst und passt sich somit nicht in ausreichendem Maße an die Bedürfnisse von technischen Implementierungen wie etwa in Unternehmen an. Technische Umsetzungen unterschiedlicher Rollenmodelle, wie etwa des ERBAC-Modells aus Kapitel 3.1.2 zeigen, dass sich in der praktischen Umsetzung eine Vielzahl von Rollen ergibt, was die Komplexität des eingesetzten Rollenmodells deutlich steigert. Eine explizite Trennung dieser beiden Sichtweisen bietet in vielfacher Hinsicht Ansatzpunkte, um ein Rollenmodell zu entwerfen, das möglichst effizient ist.

Um dieses Ziel durchzusetzen wurden in Form der Geschäfts- und Systemrollen zunächst zwei Rollen modelliert, die geschäftsnahe und technische Aspekte strikt voneinander trennen. Beide Rollen verfügen ihrerseits über eine Hierarchie, so dass voneinander unabhängige Organisationsstrukturen modelliert werden können. Zusätzlich zu den beiden Rollenbegriffen wurden generische Rollen als ein Mechanismus eingeführt, gleiche Rechte zusammenzufassen und von den konkreten Anwendungsfällen zu abstrahieren. Dadurch wird gleichzeitig das zur Verwaltung des Rollenmodells benötigte technische Wissen gesenkt, was die Akzeptanz des Modells potentiell erhöht. Eines der wichtigsten *constraints* in der rollenbasierten Zugriffskontrolle stellt *separation of duty* (SoD) dar. Im BRBAC-Modell wurde einerseits statischen SoD modelliert, mit dessen Hilfe definiert werden kann, dass sich Rollen wechselseitig ausschließen und andererseits dynamisches SoD, das den wechselseitigen Ausschluss lediglich in gewissen Kontexten anwendet. Nach diesen Modellierungen wurden Mechanismen eingeführt, die sich mit den Policys beschäftigen, die in Rollen enthalten sind und die Zugriffsrechte spezifizieren. Durch Policys mit Platzhaltern wurde ein Prinzip vorgestellt, das die Zahl an Rollen verringert. Dies wird dadurch umgesetzt, dass die konkreten Attributwerte von Policys durch Platzhalter offen gelassen und erst durch ein Benutzerattribut festgelegt werden. Im Zuge der strikten Trennung der beiden Sichtweisen ist es im Bezug auf Policys nun möglich, auf die Vererbung von Rechten einzuwirken. Dies kennt man bisher nur aus IBAC-basierten Zugriffskontrollarchitekturen. BRBAC erlaubt einerseits, die Vererbung von Rechten für eine gewisse Hierarchiestufe zu unterbinden und andererseits die Spezifikation von Rechten, die in der Hierarchie weiter unten nicht überschrieben werden dürfen. Zum Abschluss wurde durch die Modellierung von Joker-Berechtigungen ein Mehrwert dieses Rollenmodells geliefert, der automatische Rollen auf der Basis von Policys ermöglicht. Dieser Mehrwert zeigt sich speziell in der Entwurfsphase des Vorgehensmodells, welches in Kapitel 5 entwickelt wird.



Information 45: Development of the BRBAC Role Model – Complete Model

Information 45 zeigt das resultierende BRBAC-Gesamtmodell. Es definiert folgende Modellelemente und Relationen:

Modellelemente

- Benutzerkonten (engl. *user*): $U_i, \quad i \in N;$
- Berechtigungen (engl. *policy*): $P_i, \quad i \in N;$
- Rollen: $R \quad \supset BR \cup SR;$
- Kontext: $C \quad \in U_i, \quad i \in N;$
- Geschäftsrollen (engl. *business roles*): $BR_i, \quad i \in N;$
- Generische Geschäftsrollen (engl. *generic business role*):
  $BR_g \quad \subset BR;$
- Systemrollen (engl. *system roles*): $SR_i, \quad i \in N;$
- Generische Systemrollen (engl. *generic system role*):
  $SR_g \quad \subset SR;$
- Policy (engl. *policy*): $\sigma_{Pol} \quad \subseteq \quad \bigcup(ATTRIBUTE \times VALUE);$

Relationen

- Benutzer/Geschäftsrolle-Relation: $\sigma_{U,BR} \subseteq U \times BR;$
- Geschäftsrolle/Systemrolle-Relation: $\sigma_{BR,SR} \subseteq BR \times SR;$
- Geschäftsrolle/generische Geschäftsrolle-Relation (engl. *business role assignment*):
  $\sigma_{BRg,BR} \subseteq BR_g \times BR;$
- Geschäftsrolle/Systemrolle-Relation (generisch) (engl. *generic role mapping*):
  $\sigma_{BRg,SRg} \subseteq BR_g \times SR_g;$
- Systemrolle/Berechtigung-Relation: $\sigma_{SR,P} \subseteq SR \times P;$
- Systemrolle/generische Systemrolle-Relation (engl. *system role assignment*):
  $\sigma_{SRg,SR} \subseteq SR_g \times SR;$
- Statische SoD-Relation: $\sigma_{sSoD} \subseteq R \times R;$
- Dynamische SoD-Relation: $\sigma_{dSoD} \subseteq R \times R \times C;$

Funktionen

- Geschäftsrollenfunktion: *businessroles*: $U_i \rightarrow BR, \quad i \in N$;
  businessroles (u) = $\{x \mid x,y \in BR, u \in U \wedge \sigma_{U,BR}(u,x) \wedge \neg \sigma_{sSoD}(x,y) \wedge \neg \sigma_{dSoD}(x,y)\}$;
- Systemrollenfunktion: *systemroles*: $U_i \rightarrow SR, \quad i \in N$;
  systemroles (u) = $\{x \mid x,y \in SR, i,j \in BR, u \in U \wedge \sigma_{BR,SR}(i,x) \wedge \sigma_{U,BR}(u,i) \wedge \neg \sigma_{sSoD}(i,j) \wedge \neg \sigma_{dSoD}(i,j)\}$;
- Berechtigungsfunktion: *permissions*: $U_i \rightarrow P, \quad i \in N$;

Zum Abschluss dieses Kapitels sollen die in den vorangegangenen Kapiteln separat vorgestellten Anforderungen im Querschnitt betrachtet werden. Betrachtet man nun diese Mechanismen in Kombination zueinander, entstehen daraus Einsatzmöglichkeiten, die im Folgenden kurz aufgezeigt werden sollen. Es sollen an dieser Stelle lediglich Anknüpfungspunkte für weitere Untersuchungen aufgezeigt werden. Ein interessanter Aspekt etwa ist die Kombination generischer Rollen und der *wildcard*-Attribute: Generische Rollen abstrahieren bekanntlich vom konkreten Anwendungsfall oder Einsatzziel. Kombiniert man diese Rollen mit *wildcards*, lässt sich der Kontext, der im Normalfall zum Zeitpunkt der Einteilung eines Benutzers in die Rollen festgelegt werden muss, im Benutzer selbst verankern. Dies hat zur Folge, dass der Kontext nicht zum Zeitpunkt der Einteilung festgelegt wird, sondern explizit im Benutzerkonto selbst vorhanden ist. Dadurch ist es möglich, den Kontext im Wirkbetrieb bei jeder Autorisierungsanfrage zu bestimmen und daraus zur Laufzeit die für diesen Kontext gültigen Rollen zu bestimmen. Dies entlastet verständlicherweise insbesondere die Administration, da diese Aufgabe sonst manuell vorgenommen werden muss. Ein zweiter Vorteil entsteht aus einer Kombination von Geschäfts- und Systemrollen, dem wechselseitigen Ausschluss von Rollen und der Rechtevererbung: Erst durch die explizite Trennung der Rollen ist es möglich, auch technische und geschäftliche Rechte konsequent zu trennen und zwei unterschiedlichen Hierarchien zu pflegen. Dadurch ist es nun möglich, sehr feingranular auf die Rechte einzugehen und beispielsweise die Vererbung für spezielle Teilbäume zu unterbinden oder bei gewissen Rechten die Überschreibung auszuschließen. Insbesondere bieten sich die Einsatzmöglichkeiten, die aus einer Kombination der hier vorgestellten Prinzipien für weitere Forschungen an. Hier wurden zwei mögliche Vorteile skizziert, so dass an dieser Stelle durch weitere Arbeiten vertieft eingegangen werden kann.

Da in diesem Kapitel ein Rollenmodell entwickelt wurde, stellt sich nunmehr die Frage, wie vorgegangen werden sollte, um dieses Modell technisch umzusetzen. Dies wird im folgenden Kapitel näher betrachtet.

5 ENTWICKLUNG EINES VORGEHENSMODELLS FÜR DIE ROLLENBASIERTE ZUGRIFFSKONTROLLE

5.1 Zieldefinition und Festlegung der Anforderungen

In Kapitel 4 wurde mit BRBAC ein Rollenmodell mit der Zielsetzung entwickelt, geschäftliche und technische Aspekte von Rollen explizit voneinander zu trennen und dabei zu weniger komplexen Umsetzungen zu gelangen. Es stellt sich nunmehr die Frage, wie bei der Überführung dieses Rollenmodells in den Wirkbetrieb vorgegangen werden sollte. Mit genau dieser Fragestellung beschäftigt sich das vorliegende Kapitel, welches den zweiten Hauptteil dieser Arbeit darstellt. Es wird ein Vorgehensmodell entwickelt, welches sich an den klassischen Entwicklungszyklus für Software anlehnt und auf Rollen überträgt. Dazu werden in diesem Einführungskapitel zunächst zwei Ziele definiert und daraus die Anforderungen für das Vorgehensmodell abgeleitet. Abschließend werden die an diesem Prozess beteiligten Rollen eingeführt. In den anschließenden Kapiteln wird dann explizit auf die unterschiedlichen Phasen eingegangen und die hier erhobenen Anforderungen umgesetzt. Den Abschluss bildet auch in diesem Kapitel ein Resümee, welches das Modell im Gesamtkontext darstellt und die wesentlichen Punkte kurz zusammenfasst.

Das Vorgehensmodell ist in vier diskrete Phasen unterteilt, die jeweils unterschiedliche Aspekte modellieren, die bei der Umsetzung betrachtet werden müssen. Jede Phase verfolgt dabei unterschiedliche Teilziele und weist Zielartefakte auf, die in die jeweils nächste Phase übergeben werden. Da sich das Vorgehen auf das Rollenmodell aus Kapitel 4 stützt, spiegeln sich die dort definierten Ziele auch in der Vorgehensmodellierung wider. Das Ziel Z_1 des Rollenmodells BRBAC etwa erwähnt die explizite Trennung geschäftlicher und technischer Aspekte. Auch bei der technischen Umsetzung dieses Modells muss dieses Ziel aufgegriffen werden, da in den vier Phasen Mitarbeiter beteiligt sind, die über unterschiedliche Sichten auf das Unternehmen verfügen und innerhalb der Phasen unterschiedliche Teilziele realisieren. Dies führt zur Formulierung des Ziels Z_1 nach einer Differenzierung bei der Informationsakquise, die zur Umsetzung von Rollen benötigt werden. Viele Vorgehensmodelle betrachten den Entwicklungsprozess in Anlehnung an die klassische Software-Entwicklung als iterativ, der mit der finalen Abnahme durch den Kunden endet [Ba96]. Anders als klassische Software unterliegen Zugriffskontrollarchitekturen jedoch ständigen Änderungen, was Anpassungen des Rollenmodells nach sich zieht und bei dem hier entwickelten Vorgehensmodell explizit beachtet werden soll. Dies resultiert in der Definition des Ziels Z_2, welches den Lebenszyklus einer Rolle auch nach der Überführung in den Wirkbetrieb in einer eigenen Phase kapselt. Die Basis hierfür stellen Beiträge aus [KK+02], [SA+04] und [WW07] dar, in denen die Implementierung von Rollenmodellen als zyklischer Prozess dargestellt wird, der konzeptionelle Verwaltungsaufgaben modelliert, die nach der finalen Abnahme eines Rollenmodells auftreten. Auch werden mit *top-down, bottom-up* und *middle-out* drei unterschiedliche Vorgehensweisen beim Entwurf unterschieden, die sich in diesem Vorgehensmodell wiederfinden werden. Im Folgenden werden nun Anforderungen definiert und im Bezug gesetzt zu den Zielen, die sie realisieren.

Ziel Z_1: Trennung von geschäftlichen und technischen Aspekten im Vorgehen

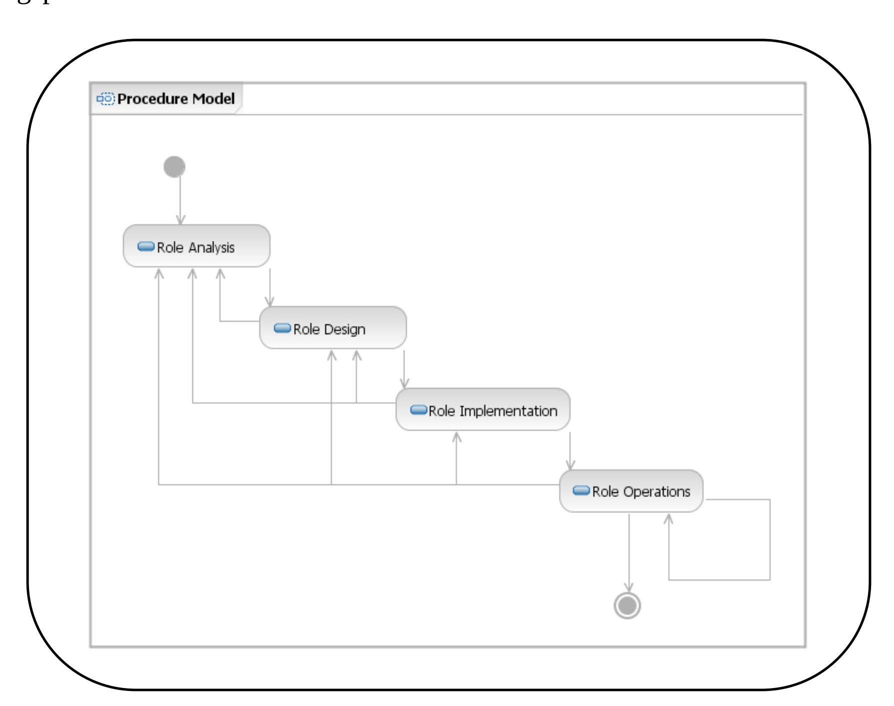
- **Anforderung $A_{1.1}$: Hybrides Vorgehen.** Wie bereits angesprochen wurde, sind an dem Prozess zur Umsetzung des Rollenmodells mehrere Mitarbeiter beteiligt, die ihrerseits teils sehr unterschiedliche Sichtweisen auf das Unternehmen haben. Das liegt daran, dass die Sichtweise auf ein Unternehmen immer durch die Beziehung geprägt ist, die man zu diesem Unternehmen hat. Ein Mitarbeiter der technischen Administration hat demnach ein radikal anderes Bild des Unternehmens, als etwa ein externer Partner, oder ein Angestellter auf Geschäftsebene. Diesem Sachverhalt wurde im Rollenmodell bereits durch die Einführung von Geschäfts- und Systemrollen Ausdruck verliehen und auch bei der technischen Umsetzung in muss dieser Tatsache Beachtung geschenkt

werden. In Anlehnung daran wird in dieser Arbeit bei der Entwicklung von Rollen auf geschäftlicher Ebene *top-down* vorgegangen und auf technischer Ebene *bottom-up*, was parallel geschieht und im Folgenden als „hybrides Vorgehen“ bezeichnet wird. Das Resultat hiervon sind die beiden in BRBAC definierten Rollentypen. Insbesondere in der Analyse- und der Entwurfsphase kommt das hybride Vorgehen zum Tragen, da in diesen beiden Phasen das Rollenmodell erzeugt wird.

Ziel Z_2: Modellierung des Vorgehens als dynamischer Lebenszyklus

- **Anforderung $A_{2.1}$: Rückkopplung bei den Vorgehensphasen.** Die Forderung nach Dynamik im Vorgehensmodell bedeutet, dass mit möglichst geringer Reaktionszeit auf sich ändernde Bedingungen eingegangen werden kann. Dies kann bei einem Vorgehensmodell entweder in den einzelnen Phasen selbst, oder über Phasengrenzen hinweg geschehen. Da das hier entwickelte Vorgehen sowohl die Entstehung des Modells mit dessen Überführung in den Wirkbetrieb, als auch die anschließenden Änderungen betrachtet, gilt es zu definieren, was zu einer Rückkopplung in eine frühere Phase führt, bzw. welche Auswirkungen eine Rückkopplung auf das weitere Vorgehen nach sich zieht.

- **Anforderung $A_{2.2}$: Explizite Modellierung eines Lebenszyklus für Rollen.** Bei der Begründung des Ziels Z_2 wurde bereits angedeutet, dass viele Vorgehensmodelle bei der Überführung eines Modells in den Wirkbetrieb enden und somit Änderungen im Wirkbetrieb nicht näher betrachten. Da Unternehmen ständig Änderungen unterliegen, ist es für ein akzeptables Vorgehensmodell nötig, dass es sich diesen Änderungen widmet und sich daran anpassen kann, sobald es sich im Wirkbetrieb befindet. Will man das Vorgehen nicht als einen Prozess betrachten, der klar endet, sondern als Lebenszyklus, der erst dann endet, wenn das Rollenmodell deprovisioniert wird, muss dies als explizite Prozessphase im Vorgehensmodell enthalten sein. In dieser Arbeit schließt sich an die erfolgreiche Umsetzung des Modells somit eine Phase an, die als Rollenverwaltungsphase bezeichnet wird.



Information 46: Procedure Model Development – Introduction

Das Vorgehensmodell ist in Information 46 dargestellt und unterteilt sich in die Phasen „Rollenanalyse“ (engl. *role analysis*), „Rollenentwurf“ (engl. *role design*) „Rollenimplementierung“

(engl. *role implementation*) und „Rollenbetrieb“ (engl. *role operations*). Für jede dieser vier Phasen wird in den folgenden Kapiteln erörtert:

- Was in dem Prozessschritt zu tun ist,
- was dessen Ziel ist bzw. welche Zielartefakte dabei entstehen,
- wer an der Phase beteiligt ist und das Voranschreiten kontrolliert und steuert und
- welche Gründe für eine Rückkopplung in einer frühere Phase sprechen könnten.

Diese Aufteilung der Kapitel wird im Folgenden konsequent verfolgt. Die in jeder Phase erzeugten Artefakte sowie mögliche Anzeichen für Rückkopplungen zu früheren Phasen bilden dabei den Abschluss jedes Teilkapitels, weil dadurch klar definiert ist, wann der Prozess in die jeweils nächste Phase eintritt, oder aufgrund von Rückkopplungen zurückgesprungen werden muss. Das Erzeugen von Artefakten ist auch deshalb von Bedeutung, weil mit dem Beginn einer neuen Phase unterschiedliche Rollen beteiligt sind und diese durch die Artefakte ein möglichst genaues Bild der bisherigen Prozessschritte erhalten sollen. Wie aus der Information 46 hervorgeht, weist das Vorgehen Rückkopplungen zu allen bereits erledigten Phasen auf, wie sie auch in aktuellen Vorgehensmodellen wie dem V-Modell oder dem Spiralmodell vorhanden sind [Ba98]. Dies macht dann Sinn, wenn in den Artefakten Schwachstellen oder Ungenauigkeiten festgestellt werden, oder evolutionäre Änderungen an der Rollenstruktur nötig sind, was etwa bei der Einführung neuer Rollen passiert. Durch die direkte Rückkopplung in alle Phasen, die bis zu diesem Zeitpunkt bereits durchlaufen wurden ist es möglich, sehr direkt in diejenige Phase zurückzuspringen, in der die Schwachstelle hervorgerufen, oder die Änderung erfasst wurde. Um dies in den folgenden Kapiteln zu verdeutlichen, wird in jeder Phase zunächst ihr Ziel definiert, anschließend, wer an der Phase beteiligt ist und wie das Ziel zu erreichen ist. Die Definition des Zielartefakts bildet den Abschluss jedes Teilkapitels. An dieser Stelle wird auch auf mögliche Schwächen eingegangen, die zu einem Rücksprung führen können.

Zum Abschluss dieses Einführungskapitels wird nun auf die Rollen eingegangen, die an dem Prozess beteiligt sind. Die Rollen stehen dabei für unterschiedliche Kompetenzen, bzw. unterschiedliche Sichten auf die Organisation, in der das Rollenmodell zu implementieren ist. Es ist eine realistische Annahme, davon auszugehen, dass diese Kompetenzen nicht nur auf verschiedene Personen aufgeteilt sind, sondern sogar auf unterschiedliche Dienstleister, die der Organisation zur Verfügung stehen. Aus diesem Grund wird die Organisation im Folgenden auch als „Kunde“ bezeichnet.

- **Die Analysten.** Die Analysten verfügen über methodische Kompetenzen und grobes technisches Wissen über das bzw. die einzusetzenden Systeme. In diesem Sinne unterteilen sich die Analysten in Business-Analyst und Sicherheits-Analyst. Sie bilden die Schnittstelle zwischen dem Kunden und den technischen Kompetenzen. Der Business-Analyst arbeitet mit der Geschäftsleitung zusammen und formalisiert deren Anforderungen an das entstehende Rollenmodell. Der Sicherheits-Analyst hat eher Kompetenzen im technischen Endsystembereich und arbeitet daher neben der Geschäftsleitung auch direkt mit den Consultants zusammen.

- **Verantwortliche Mitarbeiter beim Kunden.** Der Kunde ist in diesem Fall das Unternehmen, welches eine rollenbasierte Zugriffskontrollarchitektur einsetzen will und die explizit und implizit existierenden Geschäftsprozesse und Verantwortlichen kennt. Auch hier werden zwei Abstraktionsschichten definiert: Einerseits wird ein Zuständiger für organisatorische Strukturen benötigt und andererseits eine technische Kompetenz, der die Sichtweise der Unternehmens-IT darstellen kann.

- **Die Consultants.** Die Berater verfügen über die Kompetenzen zur technischen Umsetzung des Rollenmodells. In diesem Zusammenhang stehen die Consultants für die Schnittstelle zwischen den Analysten und den technischen Fachabteilungen des Unter-

nehmens. Sie greifen das durch die Analysten formalisierte Wissen auf und setzen es unter Einbeziehung der Fachabteilungen im Zielsystem um.

In diesem Teilkapitel wurde das Vorgehensmodell vorgestellt und dabei zunächst der Bezug zum Rollenmodell BRBAC aus Kapitel 4 vorgestellt. Anschließend wurden Ziele und Anforderungen definiert sowie die am Vorgehen beteiligten Kompetenzen vorgestellt. Im Folgenden werden die vier Phasen des Vorgehensmodells einzeln vorgestellt.

5.2 Analysephase

Die Rollenanalyse stellt den Beginn des Vorgehensmodells zur Umsetzung des Rollenmodells auf Kapitel 4 dar. In dieser Phase müssen deshalb grundlegende Fragen diskutiert werden sowie ein solides Fundament für die folgenden Phasen gelegt werden. Aus diesem Grund werden die Beteiligten an der Analyse entsprechend ihrer individuellen Sicht auf das Unternehmen unterteilt, um zu einem insgesamt geschlossenen Gesamtbild zu gelangen. Wie bereits angesprochen wurde, existieren mit *top-down*, *bottom-up* und *middle-out* drei unterschiedliche Vorgehensansätz bei der Modellierung von Rollen, die an dieser Stelle zum Verständnis kurz beschrieben werden. Die *top-down*-Variante geht dabei von einem sehr allgemeinen und abstrakt gehaltenen Bild einer Organisation aus und verfeinert dieses schrittweise, um zu geeigneten Rollen zu gelangen, wohingegen beim *bottom-up*-Vorgehen zunächst auf sehr feingranularerer Ebene begonnen wird und die dabei gefundenen Berechtigungen anschließend durch geeignete Zusammenfassungen in Rollen subsumiert werden. Der letztgenannten Ansatz beginnt bei der Definition von Rollen und versucht, diese Definition in Verbindung zur Organisation und technischen Berechtigungen zu bringen [SA+04].

Vorgehen in der Analysephase

Zu Beginn der Analysephase ist es zunächst nötig, im direkten Gespräch mit dem Kunden die Anforderungen an das entstehende Rollenmodell zu erfassen (engl. *requirement engineering*). Im Gegensatz zum Vorgehen bei klassischer Software-Entwicklung hat die Einbeziehung des Kunden in diesem Vorgehensmodell eine zusätzliche Bedeutung: Bei der Umsetzung von BRBAC sind die Anforderungen an das Resultat des Prozesses bereits klarer definiert, als bei Software. Aus diesem Grund scheint es zunächst weniger wichtig, im Rahmen einer Anforderungsanalyse die qualitativen und quantitativen Eigenschaften zu spezifizieren. Allerdings ist es möglich, dass aufgrund von Budgetbeschränkungen oder anderen Gründen zu diesem Zeitpunkt nicht unternehmensweit auf BRBAC umgestellt wird, sondern nur für einen Teil des Unternehmens. Wenn zu einem späteren Zeitpunkt die rollenbasierte Zugriffskontrolle unternehmensweit fortgeführt und integriert wird, ist diese Analyse eine wichtige Voraussetzung. Nichtsdestotrotz ist die direkte Kommunikation mit dem Kunden auch bei der Entwicklung von Rollen elementar, weil es in der Analysephase wichtig ist, das Geschäftsmodell und die Organisationsstrukturen zu erfassen. Wird BRBAC erfolgreich umgesetzt, ist davon im späteren Geschäftsablauf wenig zu spüren, weil eine Zugriffskontrollarchitektur stets im Hintergrund agiert. Durch die in BRBAC eingeführte Trennung von Geschäfts- und Systemrollen sollen geschäftliche und technische Aspekte auf einem für dieses Personal passenden Abstraktionsniveau dargestellt werden, was die Bedeutung der Anforderungsanalyse unterstreicht. Insbesondere aber die Administration, also dasjenige Personal, das am Ende des Vorgehensmodells mit dem implementierten Rollenmodell arbeitet, wird von einer klaren und strukturierten Umsetzung profitieren, was ja auch eines der Hauptziele von BRBAC ist.

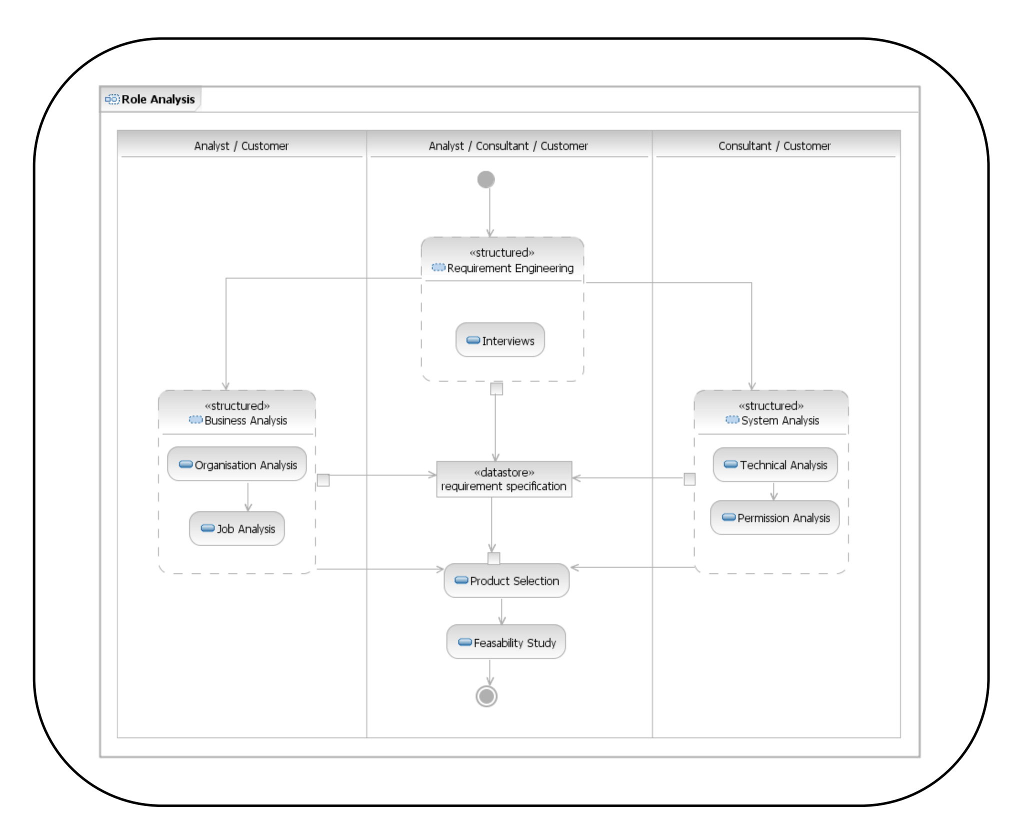
Nach der Erfassung der Anforderungen folgt die Erfassung des Ist-Zustands auf geschäftlicher und technischer Ebene. In Analogie zu [Ba96] und [WW07] steht auch in diesem Vorgehensmodell die Erfassung des Ist-Zustands im Rahmen einer Bestandsaufnahme im Vordergrund. Im Gegensatz zum Vorgehen aus [WW07], das auf dem ERBAC-Modell und dessen Erweiterungen basiert, sollen hier jedoch explizit technische und geschäftsnahe Aspekte der Rollenmodellierung unterschieden werden, weil diese Unterscheidung durch das zugrundeliegende Rollenmodell BRBAC ja explizit ermöglicht wird. Der Teilprozess zur Erfassung des Ist-Zustands ist

somit zweigeteilt und kann unabhängig voneinander geschehen. Hierdurch wird die Anforderung $A_{1,1}$ umgesetzt, die ein hybrides Vorgehen fordert.

- Auf der Geschäftsebene ist es zu diesem Zeitpunkt erforderlich, den Aufbau der Organisation und der Abteilungen zu erfassen, die im Unternehmen vorhanden sind (engl. *business analysis*). Zu dieser Beschäftigung gehört einerseits die Erfassung der verschiedenen hierarchischen Strukturen, wie etwa Abteilungen oder Niederlassungen, aber auch die Erfassung der geschäftlichen Aufgaben, die im Allgemeinen bisher weniger in formaler und eher in informeller Art definiert sind. Diese Strukturen können durch Abhängigkeitsgraphen oder die Strukturdiagramme der Modellierungssprache UML formalisiert werden [OMG07]. Zu diesem Zeitpunkt im Vorgehensmodell ist es wichtig, ein geschlossenes Gesamtbild des Unternehmens auf einer Abstraktionsstufe zu erzeugen, die für die Beteiligten intuitiv verständlich ist. Diese Erfassung beginnt somit *top-down* und die Verfeinerung – sowohl syntaktisch als auch technisch – ist erst Bestandteil der nächsten Phase.

- Die Erfassung des Ist-Zustands auf technischer Ebene (engl. *system analysis*), die unabhängig von der geschäftlichen Bestandsaufnahme geschieht, hat zum Ziel, die technischen Berechtigungen zu erfassen, die bereits vorhanden sind. Durch die Vielzahl an technischen Berechtigungen ist es im Allgemeinen sehr zeitintensiv, diese in ihrer Gesamtheit zu erfassen. Gerade im Hinblick auf systemübergreifende Rollen ist es wichtig, auf Berechtigungen zu achten, die in verschiedenen Endsystemen gleichermaßen vorhanden sind, wie etwa Systemadministratorrechte, da diese – dem *bottom-up*-Vorgehen folgend – in der nächsten Phase zu Systemrollen zusammengefasst werden können. Auch an dieser Stelle steht keine syntaktisch komplexe Erfassung im Vordergrund, sondern viel eher die konsistente Erfassung des gesamten Zustands in den technischen Endsystemen sowie der Endsysteme selbst. Dies wird in der Entwurfsphase dazu benötigt, um Systemrollen zu entwickeln. Die erfassten Endsysteme kommen dann erst in der Implementierungsphase zum Tragen, weil dort die technische Umsetzung durchgeführt wird.

Nach der Erfassung des Ist-Zustands sind die Anforderungen hinreichend spezifiziert, so dass mit der Evaluierung der Rollenmanagementwerkzeuge begonnen werden kann. In der Praxis schränken die technischen Endsysteme, die in die Implementierung des Rollenmodells aufgenommen werden sollen, die Auswahl bei den Rollenmanagementwerkzeugen bereits stark ein, weil diese nur mit bestimmten technischen Endsystemen kommunizieren können. Die Auswahl des Produktes kann an dieser Stelle dadurch eingeschränkt werden, ob die Folgenden Schritte bereits durch das Werkzeug unterstützt werden: Insbesondere die Herausarbeitung von technischen Zugriffsberechtigungen kann davon profitieren. *Role mining* (vgl. Kap 3.1.1) stellt einen praktikablen Ansatz dar, um erste Rollenhülsen aus der bestehenden Struktur herauszukristallisieren. Zu diesem Zeitpunkt steht allerdings nicht das Entwerfen von Rollen im Vordergrund, sondern die Erfassung des Ist-Zustands bei vorhandenen Berechtigungen. Dies kann die Erfassung des Ist-Zustands sowohl auf geschäftlicher- als insbesondere auch auf technischer Ebene unterstützen. Jedes Rollenmanagementwerkzeug verfügt über ein eigenes Rollenmodell, das meist auf NIST-RBAC basiert. Da dieses Modell nicht zwischen Geschäfts- und Systemrollen unterscheidet, muss die hier eingeführte explizite Unterscheidung von Geschäfts- und Systemrollen in Form einer Abbildung formalisiert werden. Die explizite Anwendung von *role mining*-Algorithmen zum Entwurf von Rollen kommt jedoch erst in der nächsten Phase zum Tragen, weil den *role-mining*-Algorithmen ein gefilterter Ausschnitt des Gesamtsystems zugeführt werden muss, auf welchem sie ausgeführt werden. Das Erfassen des Ist-Zustands stellt das Ziel dieses Teils der Analysephase dar und kann in der darauf folgenden Entwurfsphase unmittelbar zur Definition eines geeigneten Ausschnitts als Vorbedingung für den Rollenentwurf verwendet werden.



Information 47: Procedure Model Development – Role Analysis

Ziel der Analysephase

Das Ziel dieser Phase ist einerseits die Formalisierung des Gesamtzustands auf geeigneter Abstraktionsebene und andererseits die Erfassung des vom Kunden gewünschten Verhaltens. Zum Abschluss der Analysephase konzipieren die Analysten zusammen mit den Consultants einen Zeitplan sowie ein finales Lastenheft (engl. *requirement specification*), was zur Abnahme dem Kunden vorgelegt wird. Vorab wird jedoch von den Consultants zusammen mit den Analysten die Durchführbarkeit anhand des aufgestellten Zeitplans diskutiert, um über den weiteren Vorgehensverlauf zu entscheiden. Auch ist diesem Artefakt die personelle Planung beizufügen, um im Falle der Aufnahme des Projekts die nächsten Schritte und Verantwortlichkeiten zu definieren. Wie [Ba96] belegt, wird in der klassischen Software-Entwicklung die Aufwandsschätzung durch die *function point*-Methode unterstützt, die auf Erfahrungswerten aus früheren Projektumsetzungen aufbaut. Hierfür gibt es im Rollenmanagement bisher keine erkennbaren Ansätze. An dieser Stelle sei diese Methode jedoch für weitere Forschungen erwähnt. Die Abnahme durch den Kunden beschließt die Analysephase und leitet zur nächsten Phase über, dem Rollenentwurf.

Zuständigkeiten

In der Analysephase ist die Kommunikation ein elementarer Bestandteil. Da die definierten Rollen alle über eine unterschiedliche Sicht auf das Gesamtunternehmen verfügen, sollen diese Aspekte in dieser Phase alle mit eingebracht werden und das Gesamtbild ergänzen. Um den Prozess voranzutreiben, werden in der Analysephase die Zuständigkeiten hierfür definiert. Der bzw. die Zuständigen im weiteren Verlauf des Vorgehensmodells müssen für die Einhaltung eventuell abgesteckter zeitlicher Fristen sowie die Übergänge zwischen den Phasen sorgen. Die Erfassung des Ist-Zustands übernimmt die Fachabteilung zusammen mit den Beratern (engl. *consultants*), die über die Methodenkompetenz zur Erfassung des Ist-Zustands auf technischer Seite verfügen. Da die Berater direkt mit den Fachabteilungen sprechen, die ihrerseits die Verwaltung der Lösung übernehmen werden, müssen hier die gewünschten Funktionen der Lösung im Sinne eines Lastenhefts definiert werden. Die Erfassung auf geschäftlicher Ebene abstrahiert

komplett von technischen Details und wird vom Kunden zusammen mit dem Analysten übernommen. Die hier formalisierten Anforderungen fließen gleichermaßen in das Lastenheft ein.

Gründe für Rückkopplungen

In Information 47 werden die verbal beschriebenen Teilprozesse der Aktiviät Rollenanalyse (engl *role analysis*) grafisch verdeutlicht. Man erkennt hieraus, welche Schritte in dieser Phase nötig sind, wer an dem Prozess beteiligt ist und welche Artefakte dabei entstehen. Aus Information 46 sind die Rückkopplungen zwischen den Phasen aufgezeigt. Zu Rückkopplungen kann es an dieser Stelle nicht kommen, weil die Analysephase den Beginn des Vorgehens symbolisiert.

5.3 Entwurfsphase

Nach der Analyse des Ist-Zustands sowie der Spezifikation der Anforderungen in Form eines Lastenhefts werden diese Artefakte in der Entwurfsphase aufgegriffen und zu einer umsetzungsfähigen Version verfeinert. Die Durchführbarkeit wurde neben der formalisierten organisatorischen und funktionalen Struktur festgelegt und zusammen mit einzuhaltenden Fristen sowie im Prozess verantwortlichem Personal definiert. Das Ziel der Entwurfsphase ist, diese grobe Spezifikation zu präzisieren, so dass das Rollenmodell instanziiert werden kann. Dazu ist es nötig, das hybride Vorgehen fortzusetzen, um daraus Rollen abzuleiten. In analoger Weise zur Analysephase kann die Spezifizierung von Geschäfts- und Systemrollen gleichzeitig geschehen. Im Folgenden wird nun zunächst auf die Aktionen in der Entwurfsphase eingegangen, anschließend auf das Ziel der Entwurfsphase und abschließend auf die Verantwortlichkeiten sowie mögliche Gründe für Rückkopplung in die Analysephase.

Vorgehen in der Entwurfsphase

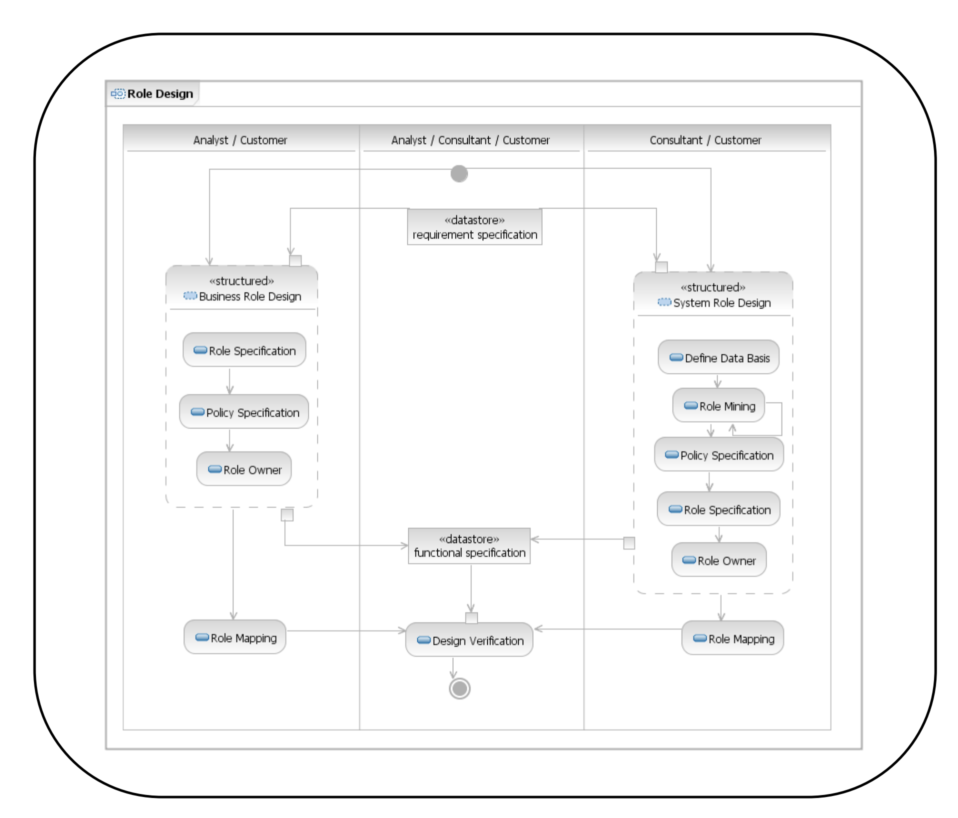
Zunächst wird die Anforderungsspezifikation, die im Lastenheft formalisiert ist, auf geschäftlicher- und technischer Ebene verfeinert. Diese Aufteilung verringert die Komplexität in diesen beiden Bereichen und fokussiert für jeden Bereich auf den hierfür wesentlichen Ausschnitt des Unternehmens. Auf geschäftlicher Ebene steht somit im Mittelpunkt, für die bisher formalisierten Abteilungen und Geschäftsprozesse Geschäftsrollen zu definieren und eine Hierarchie zu erfassen (engl. *business role design*). Dieser Teil des hybriden Vorgehens setzt die in der Analysephase begonnene *top-down*-Modellierung fort, da ausgehend von den allgemein gehaltenen Strukturen schrittweise verfeinert wird. In der technischen Spezifikation wird hingegen das *bottom-up* Vorgehen aus der ersten Phase fortgeführt, weil die in der Analysephase vorhandenen technischen Berechtigungen nun zu Systemrollen zusammengefasst werden sollen (engl. *system role design*). An dieser Stelle ist eine Werkzeugunterstützung in Form von *role-mining*-Algorithmen von Vorteil, da mit deren Hilfe aus dem Datenbestand des Unternehmens Rollen automatisiert abgeleitet werden können. Wie in Kapitel 3.1.1 bereits angesprochen wurde, muss vor dem Anwenden dieser Algorithmen das Gesamtsystem auf einen geeigneten Ausschnitt reduziert werden. Dazu dient die Formalisierung des Ist-Zustands auf technischer Ebene aus dem Lastenheft, die unter Einbeziehung der Fachabteilungen entstanden ist. Aufgrund der Verifikation dieser Formalisierung durch den Kunden am Ende der Analysephase kann man davon ausgehen, dass der erfasste Ausschnitt für das zu implementierende Rollenmodell angemessen spezifiziert wurde. Durch schrittweises Verfeinern des Datenbestands können die Systemberechtigungen nun iterativ zusammengefasst werden, um daraus Rollen abzuleiten, die für den jeweiligen Einsatz im Unternehmen angemessen sind. Durch diese Tätigkeit werden die Formalisierungen aus der Analysephase somit einerseits zu Rollen weiterentwickelt und andererseits der Berechtigungsumfang für diese Rollen definiert.

Am Ende der Spezifikation beider Rollentypen muss für jede Rolle ein Besitzer (engl. *owner*) definiert werden, der sich in der späteren Betriebsphase für diese Rolle verantwortlich zeigt. Hierbei ist insbesondere wichtig, dass dieser Verantwortliche auch die Struktur der Systemrollen kennt, weil er in der Rollenbetriebsphase neben der Pflege und der Einteilung von Benutzern insbesondere auch die Einteilung zu Systemrollen durchführen wird. In der Regel kann es einen Verantwortlichen auf Geschäftsebene geben, der einen Rollenverantwortlichen anweist, die er-

wähnten Änderungen durchzuführen. Diese explizite Unterscheidung ist an dieser Stelle allerdings von untergeordneter Bedeutung. Die Einbeziehung der Fachabteilungen und das hohe Maß an interner Kommunikation erweist sich hierbei als elementarer Bestandteil des Gesamtprozesses. Da die bisherigen Definitionen der System- und Geschäftsrollen unabhängig voneinander geschehen, ist es im der Folge notwendig, den Bezug zwischen den Rollentypen zu definieren und die Rollenverantwortlichen in die gewählte Spezifikation einzuführen.

Diese Tätigkeit ist daher mit großer Sorgfalt durchzuführen und kann erst dann vorgenommen werden, wenn die Verantwortlichkeiten geklärt sind. An dieser Stelle zeigt sich ein Vorteil der Trennung von geschäftlichen und technischen Belangen, weil ein Geschäftsrollenverantwortlicher keine Kenntnis über technische Details, sondern lediglich über die Bedeutung einer Systemrolle kennen und verstehen muss. Ein Systemrollenverantwortlicher auf der anderen Seite benötigt keine detaillierten Kenntnisse über die Struktur von Geschäftsrollen und kann sich auf die Pflege von systemabhängigen Berechtigungen und Policys konzentrieren. Im Folgenden muss nun von den Rollenverantwortlichen die Verknüpfung von Geschäfts- und Systemrollen bewerkstelligt werden. Obwohl dieses Personal erst in der Rollenbetriebsphase mit dem Rollenmanagementwerkzeug in Kontakt kommen wird, ist es aus den genannten Gründen von Vorteil, sie in dieser Phase ebenfalls mit einzubeziehen, weil sie so aktiv an der Gestaltung beteiligt sind und ihre individuelle Sichtweise des Gesamtsystems in den Entwurf mit einbringen können.

Im Zuge des Rollenentwurfs wird das Lastenheft zu einem Pflichtenheft spezifiziert. Diese Spezifizierung hat das Ziel, die formal definierten Anforderungen und Modellierungsaspekte in eine Form zu übertragen, die in der Implementierungsphase für das dort zuständige Personal unmittelbar verstanden werden kann. Dazu ist es beispielsweise nötig, die bisher sehr abstrakt formalisierten Sachverhalte durch geeignete Modelle zu konkretisieren. Dies umfasst die Spezifikation des einzusetzenden Rollenmodells, aber auch die funktionale Spezifikation. Letztere etwa beinhaltet unter anderem sämtliche administrative Funktionen, die für den Wirkbetrieb des Rollenmodells benötigt werden, oder Leistungsindikatoren, an denen die Qualität der technischen Umsetzung gemessen werden kann. Das Pflichtenheft erfüllt insbesondere für den Kunden noch eine zusätzliche Aufgabe: In Projekten dienen die Spezifikationen im Pflichtenheft als rechtliche Grundlage für die Umsetzung des Modells in den Wirkbetrieb und legen daher das Ende der Implementierungsphase fest. Aus diesem Grund wird das Pflichtenheft als Artefakt am Ende der Entwurfsphase dem Kunden zur gewissenhaften Überprüfung vorgelegt. Dies sollte ohne Einbeziehung der Analysten oder Consultants geschehen, damit sich die Verantwortlichen ohne Einflüsse von Außen mit dem zu implementierenden Modell auseinandersetzen und es diskutieren können.



Information 48: Procedure Model Development – Role Design

Ziel der Entwurfsphase

Das Ziel der Entwurfsphase ist es, zu einem ausspezifizierten Rollenmodell zu gelangen, welches in der Implementierungsphase umgesetzt werden kann. Dazu gehören ausspezifizierte Modelle der beteiligten Rollen, Geschäfts- und Systemrollen und den Berechtigungen in den Endsystemen (engl. *functional specification*). Durch die Definition von Rollenverantwortlichen ist für die Implementierungsphase auch schon ein Teil der Umsetzungsverantwortung dezentral definiert. Durch die Beteiligung der Verantwortlichen am Entwurf ist sichergestellt, dass die geplante Umsetzung auch in deren Sinne erfolgt. Durch die finale Abnahme durch den Kunden wird sichergestellt, dass das bisher erarbeitete in seinem Interesse umgesetzt wird (engl. *design verification*). An diesem Gespräch müssen Consultants und Analysten nicht unbedingt anwesend sein, da dies die Möglichkeit einräumt, dass die Verantwortlichen des Unternehmens im direkten Gespräch miteinander eventuell als kritisch erachtete Punkte des Entwurfs entdeckt und diskutiert. Ein abschließendes Interview mit Consultants und Analysten, in dem der Kunde die Ergebnisse der Verifikation vorträgt, ist angedacht und beschließt damit das Ende der Entwurfsphase. Alle Aktionen und Zuständigkeiten sind in Information 48 dargestellt. Am Ende der Entwurfsphase sind die unterschiedlichen Rollen samt Berechtigungen und Benutzern hinreichend spezifiziert und können nach der finalen Abnahmen durch den Kunden in einem Rollenmanagementwerkzeug implementiert werden. Dies ist Bestandteil der Implementierungsphase.

Zuständigkeit

Die Entwurfsphase selbst ist durch die Komplexität des Gesamtsystems sowie die unterschiedlichen Kompetenzen sehr arbeitsteilig und erfordert eine ebenso aktive Kommunikation zwischen allen Beteiligten. Der *top-down*-Entwurf auf Ebene der Geschäftsleitung wird dabei maßgeblich geprägt von den Analysten, die in der Analysephase durch die Gespräche mit dem Kunden in die Organisation sowie die formalisierten Aufgaben innerhalb des Unternehmens eingeführt wurden. Aus diesem Grund verfügen Sie über den angemessen Ausschnitt der Unternehmensstruktur und können die Geschäftsrollen sowie deren Hierarchie zusammen mit geschäftlichen Aufgaben in Form von Geschäfts-Policys definieren.

Der *bottom-up*-Ansatz, der auf der Ebene der technischen Systemen verfolgt wird, entsteht unter Beteiligung der Consultants, die zusammen mit den Fachabteilungen in einem iterativen Vorgehen die technischen Berechtigungen aus der Analysephase nach Zugriffsmustern durchsuchen und zu Systemrollen kapseln. Wie bereits angesprochen wurde, kann dieser Prozess durch geeignete Werkzeuge zum automatisierten Entwickeln von Rollen aus dem Datenbestand des Unternehmens (engl. *role mining*) unterstützt werden. Andernfalls gestaltet sich diese Phase als sehr arbeitsintensiv, weil es im Allgemeinen sehr viele Einzelberechtigungen gibt, die individuell konsolidiert werden müssen, um eine konsistente Implementierung dieser Rollen in der nächsten Phase zu gewährleisten. Durch die Einbeziehung der Geschäftsleitung sowie der Fachabteilungen wird das Verständnis des entstehenden Rollenmodells stark gefördert, weil sie sich selbst intensiv einbringen können und das implementierte Rollenmodell dadurch stark beeinflussen können. Auf der anderen Seite ist das auch deshalb notwendig, weil die Consultants ohne das Feedback der Kunden das Unternehmen nicht in geeignetem Maße auf das Rollenmodell abbilden können, weil ihnen zumindest das implizite Wissen über Abläufe im Unternehmen fehlt. Durch diesen Ansatz werden kleinere Schwachstellen, die in der Analysephase zu unpräzise oder gar nicht beachtet wurden, umgehend behoben. Bei größeren Versäumnissen, die zu großen Änderungen des bisherigen Modells führen würden, ist es ratsam, in die Analysephase zurückzukehren, weil dies ein Indiz für eine unzureichende Analyse ist. Darauf wird im Folgenden noch näher eingegangen. Am Ende dieser Phase wird das resultierende Rollenmodell, zusammen mit Geschäfts- und Systemrollen sowie den darin enthaltenen Berechtigungen und Benutzern und dem Pflichtenheft dem Kunden übergeben, der dieses dann zu verifizieren hat.

Gründe für Rückkopplungen

Zu Rückkopplungen kann es in dieser Phase dann kommen, wenn bei der finalen Abnahme durch den Kunden festgestellt wird, dass gewisse Charakteristika der zu implementierenden Lösung im Pflichtenheft nicht vorhanden sind. Zwar ist die Auftretenswahrscheinlichkeit für Fehler im Pflichtenheft durch die Anforderung $A_{1,1}$ nach einem hybriden Vorgehen mit starker Einbeziehung der Fachabteilungen sehr gering, aber Fehler können dennoch nie ganz ausgeschlossen werden. In diesem Fall muss in die Analysephase zurückgekehrt werden und sowohl die personellen Ressourcen, als auch die zeitlichen Fristen, die für das Vorgehensmodell veranschlagt wurden, überarbeitet werden. Ein zweites Indiz für Rückkopplung stellt der Systemrollen-Entwurf dar: Sollten bei der Anwendung der *role-mining*-Algorithmen keine geeigneten Systemrollen gefunden werden können, ist der zugrundeliegende Datenbestand möglicherweise nicht umfassend genug und deutet somit auf einen Fehler in der Analysephase hin, da der Datenbestand dort definiert wurde. Ein drittes Indiz für eine Rückkopplung ist, wenn der Kunde zum Abschluss des Entwurfs mit hierbei festgelegten Entwurfsentscheidungen nicht einverstanden ist. Diese Bedenken können im erwähnten Interview mit Analysten und Consultants entweder ausgeräumt, oder unmittelbar nachgebessert werden. Bei größeren Nachbesserungen sollte allerdings auch hier explizit in die Analysephase zurückgekehrt werden, um das Gewünschte nachzumodellieren.

5.4 Implementierungsphase

In Übereinstimmung mit dem Vorgehensmodell aus [WW07] beschreibt die Implementierungsphase für Rollen denjenigen Teilprozess, in dem das spezifizierte Rollenmodell in den Wirkbetrieb überführt wird und die entworfenen Rollen in der Gesamtheit technisch umgesetzt werden. Die Grundlage hierfür bilden die Artefakte aus der Entwurfsphase, nachdem sie durch den Kunden abgenommen und für gut befunden wurden. Dieser Teilprozess ist in zwei Phasen unterteilt: Zunächst muss das Rollenmanagementwerkzeug selbst im Unternehmen verteilt werden, um anschließend das Rollenmodell auf dieses übertragen zu können. Auf diese beiden Ziele wird im Folgenden eingegangen und erläutert, wer an diesen Phasen beteiligt ist und was gegebenenfalls zu Rückkopplungen in die Entwurfs- oder Analysephase führen kann.

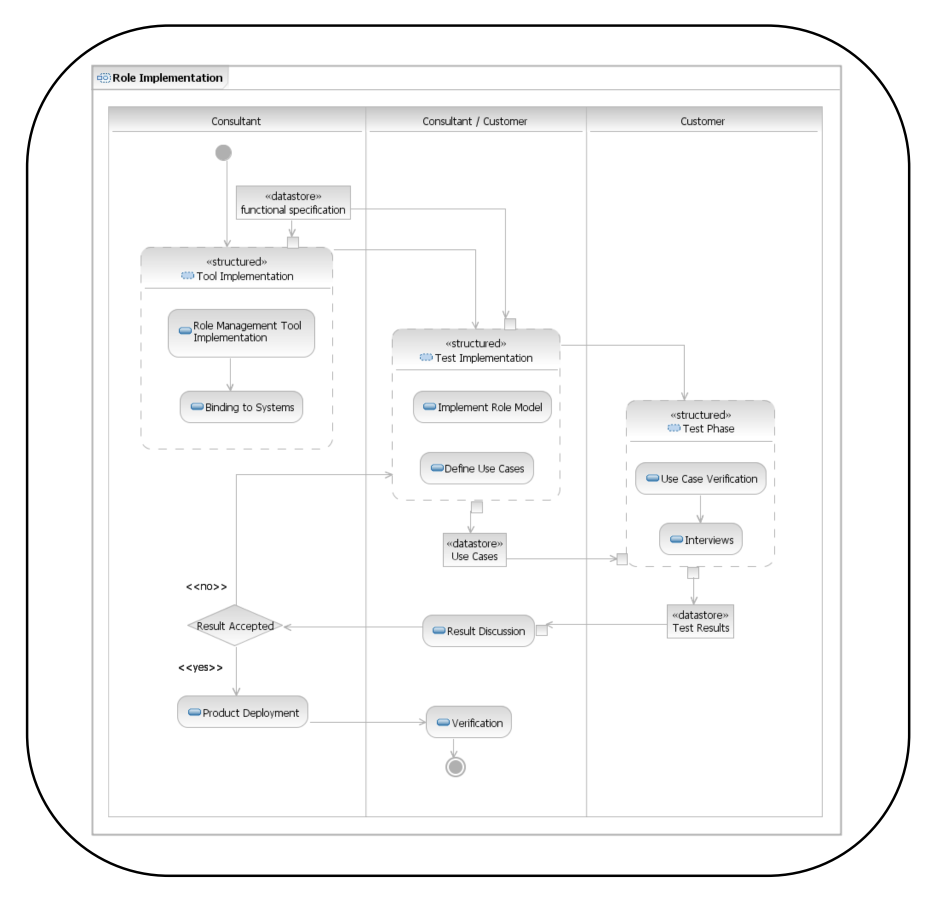
Vorgehen in der Implementierungsphase

Zu Beginn der Implementierungsphase ist es zunächst nötig, das Rollenmanagementwerkzeug im Unternehmen zu implementieren und die Verbindung zu den beteiligten Endsystemen herzustellen (engl. *tool implementation*). Diese Tätigkeit hat noch keinen expliziten Bezug zu den Artefakten aus der Entwurfsphase, da es dort vorrangig um den Entwurf des Rollenmodells geht. Jedoch wurde bereits in der Analysephase im Rahmen der Bestandsaufnahme auch der Ist-Zustand auf technischer Ebene erfasst. Hieraus geht nun unter anderem hervor, welche Endsysteme im Unternehmen vorhanden und dementsprechend in die Zugriffskontrollarchitektur eingebunden werden müssen. Zum Teil kann dies bereits umgesetzt worden sein, wenn *role mining*-Algorithmen (vgl. Kap 3.1.1) von dem Werkzeug angeboten werden und diese zum Rollenentwurf herangezogen wurden. Zur vollständigen Einbindung werden Verbindungen zwischen dem Rollenmanagementwerkzeug und den erwähnten Endsystemen hergestellt, aus denen die in der Entwurfsphase spezifizierten Daten für die rollenbasierte Zugriffskontrolle bezogen werden sollen. An dieser Stelle sei erwähnt, dass zunächst ein Testbetrieb des Werkzeugs für eine ausgewählte Abteilung angestrebt wird, was bei der Spezifizierung der Verbindung zu den Endsystemen beachtet werden muss, um den Gesamtdatenbestand durch den Testbetrieb nicht zu kompromittieren oder ungewollt zu verändern (engl. *test implementation*). Für die Dauer der Testphase bleibt die bisher eingesetzte Zugriffskontrolle autoritativ, während BRBAC im Testbetrieb lediglich Daten von dort bezieht und redundant vorhält. Auch ist ein Szenario denkbar, in dem keine unternehmensweite Umstellung auf BRBAC, sondern ein Parallelbetrieb mehrerer Zugriffskontrollarchitekturen angestrebt wird.

Nachdem das Werkzeug erfolgreich implementiert und die Kommunikationsbeziehung zu den Endsystemen umgesetzt wurde, wird das Rollenmodell gemäß der Spezifikationen aus der Entwurfsphase nun in den Wirkbetrieb überführt. In der Entwurfsphase kamen bereits *role mining*-Algorithmen zum Einsatz, um die Spezifikation von Rollen zu unterstützen. Eine planmäßige Einführung von Geschäfts- und Systemrollen findet allerdings erst jetzt in der Implementierungsphase statt, was damit begründet werden kann, dass erst nach Vervollständigung der Artefakte das Gesamtbild des Unternehmens spezifiziert ist. Zur vollständigen Einführung von BRBAC werden die Artefakte der Entwurfsphase mit den bisher erzeugten Rollenhülsen zusammengeführt und technisch in der Rollenmanagementlösung umgesetzt. Für den Fall, dass das eingesetzte Werkzeug Kompetenzen aus dem *role engineering*, wie etwa *role mining*-Techniken verfügt, kann die Umsetzung der modellierten Rollen auf das Werkzeug stark vereinfacht werden. Andernfalls müssen die Rollenimplementierungen manuell vorgenommen werden. Da wie bereits erwähnt, zunächst ein eingeschränkter Testbetrieb angestrebt wird, hält sich der zeitliche Aufwand hierfür jedoch auch für größere Einrichtungen in einem überschaubaren Rahmen, wie auch [WW07] belegt. Nachdem das Rollenmodell für die ausgewählte Abteilung überführt wurde, ist es wichtig, diese Benutzer entsprechend einzuführen. Der Sinn davon ist, dass die Benutzer zunächst über den Testbetrieb selbst und anschließend über die in diesem Rahmen von ihnen zu erledigenden Aufgaben informiert sind.

Wie $A_{2,1}$ fordert, sollen Erfahrungswerte im Vorgehensmodell möglichst direkt verarbeitet werden können. Aus diesem Grund macht es Sinn, in einer ausgewählten Abteilung zunächst einen Parallelbetrieb zu ermöglichen und erst nach der Beendigung dieser Testphase auf das gesamte Unternehmen auszuweiten (engl. *test phase*). Die während der Testphase gesammelten Erfahrungswerte im Umgang mit dem implementierten Rollenmodell können dann zur Nachbesserung in dieses zurückgeführt werden. Allerdings kann dies erst dann geschehen, wenn das Rollenmanagementwerkzeug in den Testbetrieb überführt wurde und das Rollenmodell für die ausgewählte Testabteilung implementiert wurde. Vor dem Beginn dieser Testphase müssen Testfälle aufgestellt und den Mitarbeitern vorgestellt werden, die sich aus den Anforderungen aus dem Pflichtenheft ableiten lassen. Dieses Vorgehen liegt in sofern nahe, als dass das Pflichtenheft die Testfälle bereits in formaler Weise enthält. In dieser Phase ist es wichtig, die Leistung der Pilotierung zu überprüfen, etwa um frühzeitig Schwachstellen oder Leistungsengpässe zu erkennen. Am Ende dieser Testfälle und der Erfassung von Schwächen werden die Testbenutzer intensiv interviewt, um Erfahrungswerte in der Benutzung zu erfassen. Das Ergebnis die-

ser direkten Gespräche ist eine Liste mit offenen Punkten, die vor einem Ausrollen des Rollenmodells auf das Unternehmen abgearbeitet werden müssen.



Information 49: Procedure Model Development – Role Implementation

Zuständigkeiten

Die Implementierungsphase ist stark getrieben durch die Consultants, die zunächst dafür verantwortlich sind, das Rollenmanagementwerkzeug im Unternehmen einzusetzen und im Anschluss daran die Verbindung zum Unternehmensdatenbestand herzustellen. Die Überführung des Rollenmodells in den Wirkbetrieb ist ebenfalls Aufgabe der Consultants. Es kann durchaus Sinn machen, die technische Kompetenz im Unternehmen – vertreten durch die Fachabteilungen – an diesem Prozess aktiv zu beteiligen, da so der Wissenstransfer unterstützt wird, was in der Folge zu einer guten Verwaltung des Rollenmodells durch die Fachabteilungen führt und somit der nächsten Phase zugute kommt. Nachdem sich das Rollenmodell im Testbetrieb befindet, beginnt ein Teilprozess, an dem wiederum die Consultants zusammen mit der technischen Kompetenz beteiligt sind und Testfälle auf der Basis des Pflichtenhefts definieren. Zusätzlich dazu werden die Testfälle mit Leistungsindikatoren versehen, die während des Testbetriebs überwacht werden können. Die Überwachung selbst kann dann bereits ohne explizite Einbeziehung der Consultants vom Kunden selbst durchgeführt werden. Am Ende der Testphase werden, wie bereits angesprochen wurde, Interviews mit den Testbenutzern durchgeführt, die von den Consultants und der technischen Fachabteilung durchgeführt werden. Durch diese Interviews sollen Erfahrungen erfasst werden, die sich aus den reinen Leistungsindikatoren nicht ablesen lassen. Dies können etwa Schwachstellen in den abgebildeten Prozessen oder subjektive Eindrücke bei der Durchführung typischer Aufgaben sein.

Ziel der Implementierungsphase

Das klare Ziel dieser Phase ist, das entworfene Rollenmodell in den Wirkbetrieb zu überführen und dabei zunächst eine Testphase zu pilotieren. Zu diesem Zeitpunkt ist die bisher im Einsatz

befindliche Zugriffskontrollarchitektur weiterhin autoritativ und die rollenbasierte Architektur lediglich Dienstnehmer. Nach erfolgreichem Testbetrieb wird BRBAC dann Schritt für Schritt auf das gesamte Unternehmen ausgeweitet, wobei immer noch ein Parallelbetrieb verfolgt wird. Durch dieses Vorgehen können schrittweise weitere Erfahrungen aus unterschiedlichen Abteilungen gesammelt werden, die eine falsche Benutzung der Rollenmanagementlösung, oder eine inkorrekte Modellierung der Rollen sehr unwahrscheinlich werden lässt. Nachdem der Parallelbetrieb von BRBAC auf das gesamten Unternehmen ausgeweitet wurde und einer finalen Überprüfung der Erfahrungswerte aus der bisherigen Umsetzung stand hält, kann BRBAC zur autoritativen Zugriffskontrollarchitektur im Unternehmen werden und das Altsystem ersetzt werden. Das Ende der Implementierungsphase wird durch die Abnahme durch den Kunden signalisiert. Anhand des Pflichtenhefts sowie der Leistungsindikatorergebnisse aus dem Testbetrieb wird diese Abnahme durchgeführt. Dieses Vorgehen ermöglicht einen sehr sanften Umstieg für die einzelnen Abteilungen und einen effektiven Umgang mit der neuen Technologie für das gesamte Unternehmen. Speziell der sich schrittweise ausweitende Testbetrieb ist allerdings gerade für größere Einrichtungen ressourcenintensiv. Einerseits ist es personal- und zeitintensiv, aber auch die technischen Ressourcen für den Parallelbetrieb müssen zur Verfügung stehen. Es ist daher auch möglich, das Ausrollen des Rollenmodells nach einem pilotierten Testbetrieb direkt auf das gesamte Unternehmen auszudehnen, jedoch leidet darunter möglicherweise die Akzeptanz des Rollenmodells bei den Angestellten. Der gesamte Implementierungsprozess ist in Information 49 grafisch dargestellt.

Gründe für Rückkopplungen

Wie aus der Abbildung erkennbar ist, ist die Implementierungsphase unterteilt in einen Implementierungsteil und einen Testbetrieb. In der Testbetriebphase zeigt sich, wie sorgfältig in der Rollenentwurfsphase modelliert wurde, da durch die Testfälle sowie die anschließenden Interviews mit den Testkandidaten die Resonanz bei den Angestellten erfasst wird. Je konzentrierter die finalen Artefakte in der Entwurfsphase spezifiziert wurden, umso geringer ist die Wahrscheinlichkeit, dass durch einen Rücksprung in die Entwurfs- oder gar die Analysephase nachgebessert werden muss. In Absprache mit dem Kunden kann es somit an dieser Stelle zur Überarbeitung einer vorhandenen Anforderung, oder gar einen Rücksprung in die Entwurfs- oder Analysephase kommen. In diesem Fall muss der Zeitplan für die Umsetzung überarbeitet werden, weil durch Rückkopplungen der ursprünglich anvisierte Zeitplan unter Umständen nicht eingehalten werden kann. Bei der Ausweitung des Parallelbetriebs auf das gesamte Unternehmen werden zwei Ziele verfolgt, die ihrerseits zu Rückkopplungen führen können: Einerseits können bei der Ausweitung neue Erkenntnisse hinzukommen, die eine Nachbesserung des Modells nach sich ziehen. Andererseits kann Nachbesserungsbedarf seitens der Benutzer bestehen, der auf Bedienschwierigkeiten zurückgeführt und somit unmittelbarer behoben werden kann, als es bei tiefgreifenden Versäumnissen im Rollenmodell der Fall ist. Durch den Parallelbetrieb jedoch wird ja insbesondere das Ziel verfolgt, dass sich die Benutzer an die neue Zugriffskontrolle gewöhnen. Die Liste der Nachbesserungswünsche gilt gleichermaßen als Prüfstein für das umgesetzte Rollenmodell: Kommt es zu vielen Problemen oder Engpässen bei den Leistungsindikatoren, ist dies ein deutliches Indiz dafür, dass in der Entwurfsphase nicht sorgfältig genug modelliert wurde und somit in die Entwurfsphase zurückgekehrt werden sollte.

5.5 Betriebsphase

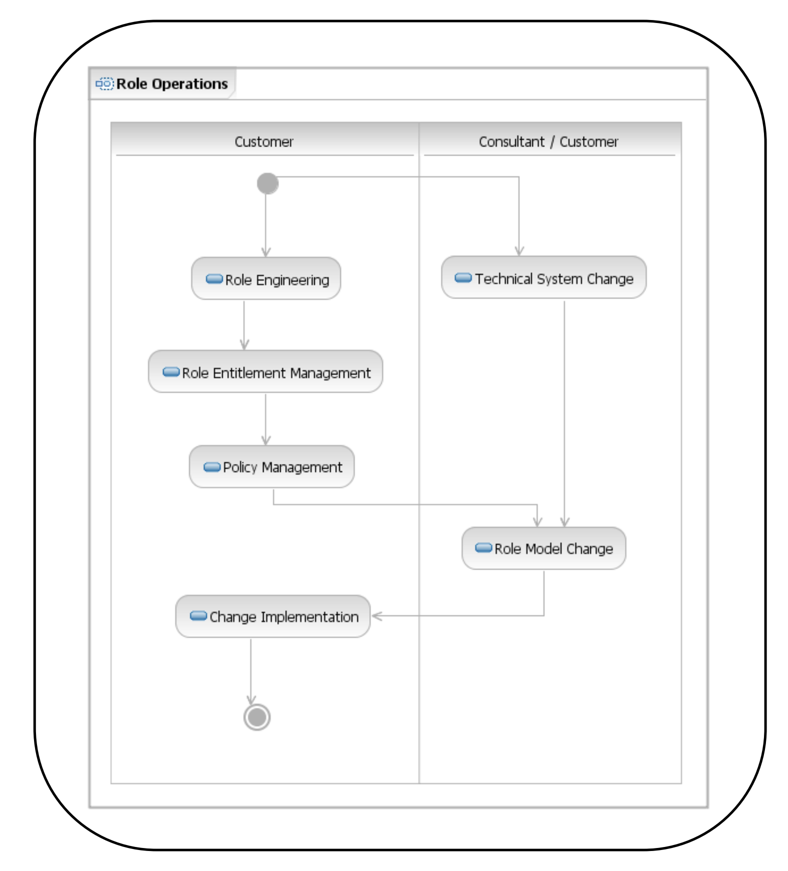
Der zentrale Aspekt dieser Phase ist die Pflege des Rollenmodells im Wirkbetrieb. Anders als bei klassischer Software unterliegt die Facharchitektur bzw. die zugrundeliegende Zugriffskontrollarchitektur fortwährend Änderungen. Wie aus Information 46 bereits hervorgeht, weist diese Phase als einzige eine Rückkopplung auf sich selbst auf, womit ausgedrückt werden soll, dass die Betriebsphase eine fortlaufende Phase im Lebenszyklus des Rollenmodells darstellt, wie in $A_{2,2}$ gefordert wurde.

Vorgehen in der Betriebsphase

Diese Phase richtet sich an die technische Fachabteilung in gleichem Maße, wie an die Rollenverantwortlichen im Unternehmen. Diese arbeiten nun an dem Rollenmanagementwerkzeug zur Verwaltung von Rollen, Berechtigungen und Policys auf unterschiedlichen Ebenen und mit unterschiedlichem Detailverständnis. Die Änderungen, die sich in dieser Phase ergeben, können dabei grob in zwei Arten unterteilt werden: Änderungen, die das Rollenmodell selbst betreffen sowie Änderungen der technischen Endsysteme.

Der erstgenannte Fall beschreibt Änderungen an den Zuweisungen von Rollen, Benutzern oder Policys und Änderungen am Rollenmodell (engl. *role engineering*, *role entitlement management*, *policy management*). Zuweisungsanpassungen treten dabei wesentlich häufiger auf, da das Hinzufügen von Benutzern, das Entfernen oder Ändern über das gesamte Unternehmen betrachtet sehr häufig passieren. Dies sind administrative Tätigkeiten, die vom eingesetzten Rollenmanagementwerkzeug in automatischer Weise bzw. von den technischen Fachabteilungen oder den Rollenverantwortlichen in manueller Weise durchgeführt werden. Die starke Einbeziehung und aktive Beteiligung dieses Personals in der Entwurfsphase stellt jedoch sicher, dass sie über die nötigen Fähigkeiten zur Erledigung dieser Aufgaben verfügen. Rollenmodellanpassungen (engl. *role model changes*) hingegen betreffen das Schema des Rollenmodells, dessen Strukturen oder die darin definierte Abbildungen. Änderungen etwa an der Rollenstruktur oder den Hierarchien sind tiefgreifendere Änderungen, wie sie etwa beim Zusammenlegen von Abteilungen, oder Zukäufen von Firmen der Fall sind. Es wird empfohlen, an dieser Stelle die technischen Berater (engl. *consultants*) einzubeziehen, weil diese einerseits den bisherigen Prozess begleitet haben und darüber hinaus über das notwenige technische Wissen verfügen.

Der zweite Fall betrifft Änderungen an den technischen Endsystemen selbst (engl. *technical system change*), etwa, wenn Systeme aus der Verwaltung durch das Rollenmanagementwerkzeug ausgegliedert werden, oder hinzukommen. In diesem Fall, der für die Praxis auch sehr relevant ist, ist es zunächst wichtig, die Änderungen zu formalisieren, die sich aus einer Hinzunahme oder einem Entfernen von technischen Endsystemen ergeben. Beim Hinzufügen muss zumindest auf technischer Ebene erneut in die Analysephase zurückgesprungen werden, um die in diesen Systemen vorhandenen Berechtigungen zu erfassen und diese im Anschluss daran zu Systemrollen zusammenzufassen. Durch die Trennung von Geschäfts- und Systemrollen kann es allerdings verhindert werden, dass das Vorgehen auch für die geschäftliche Ebene in der Analysephase startet, weil in BRBAC die technischen Systeme von der Geschäftslogik entkoppelt sind. Beim Entfernen eines oder mehrerer Endsysteme muss erfasst werden, welche Systemrollen den Zugriff auf diese steuern um diese in planvoller Weise deprovisionieren zu können. Auch hierbei kann die geschäftliche Ebene in gleicher Weise übernommen werden. In diesem Fall kann man somit von einer gemischten Rückkopplung sprechen.



Information 50: Procedure Model Development – Role Operations

Ziel der Betriebsphase

Diese stark zyklische Phase hat als Ziel die Sicherung des Wirkbetriebs, die Erhaltung der Konsistenz des Rollenmodells sowie damit einhergehend die gesetzlichen und technischen Policys des Unternehmens dauerhaft einzuhalten. Aus diesem Grund gibt es kein klar definierbares Ende dieser Phase. Die Bezeichnung „Lebenszyklus", die in Anforderung $A_{2.2}$ angesprochen wird, soll symbolisieren, dass das Ende dieser Phase genau dann erreicht wird, wenn das Rollenmodell aus dem Wirkbetrieb entfernt wird, oder durch eine planmäßige Änderung des Modells in eine der davor liegenden Phasen zurückgesprungen wird. Die Konsistenz der Daten soll dabei stets gewahrt bleiben. Das Ende eines Rollenmodells kann etwa dann erreicht sein, wenn eine andere Architektur als autorative Quelle für die Zugriffskontrolle eingesetzt werden soll. Das am Rollenbetrieb beteiligte Personal sowie die hierbei zu erledigenden Teilprozesse sind in Information 50 dargestellt.

Gründe für Rückkopplungen

Um Änderung im Vorgehen möglichst direkt zu unterstützen, kann in den einzelnen Phasen in alle zeitlich davorliegenden Phasen zurückgesprungen werden, sobald die Fehlerquelle oder der Ansatzpunkt der Änderung identifiziert wurde. Dabei wird stets in die Phase zurückgesprungen, die vor der Phase liegt, in der die Änderung hervorgerufen wurde. Die einzige Phase, die eine Rückkopplung auf sich selbst aufweist, ist die hier definierte Wartungsphase, wodurch ausgedrückt werden soll, dass sie einen fortwährenden Prozess beschreibt, der nur bei geplanten Änderungen am Rollenmodell oder dessen Terminierung verlassen wird. Eine kurze Taktung bei der Überprüfung der in den Phasen definierten Teilziele ist notwendig, um Abweichungen vom Vorgehen, oder geplante Änderungen möglichst zeitnah zu erkennen. Wie bereits erwähnt wurde, kommt es in dieser Phase praktisch fortlaufend zu Änderungen, weil sich am Unternehmensdatenbestand wie etwa der Rollenzuordnungen, den Benutzern oder den technischen Berechtigungen sehr häufig Änderungen ergeben. Das können etwa neue Mitarbeiter sein, die in das Unternehmen eintreten, oder durch eine Änderung ihrer Position über andere Rechte verfü-

gen. Die gemischte Rückkopplung, die für einen Rücksprung in die Analyse- aber auch die Entwurfsphase steht, beschreibt eine besondere Form der Rückkopplung und tritt auf, wenn sich an den zugrundeliegenden technischen Systemen Änderungen ergeben. Hierbei muss analysiert werden, welche Änderungen auf geschäftlicher, insbesondere aber auf technischer Ebene anfallen. Da dies ein sehr relevanter Aspekt für die Praxis darstellt, bietet er sich für weitere Untersuchungen im BRBAC-Kontext an.

5.6 Resümee

In diesem Kapitel wurde ein Vorgehensmodell entwickelt, welches sich auf das Rollenmodell BRBAC stützt, was in Kapitel 4 entwickelt wurde. Dazu wurden vier Prozessphasen definiert, die vom klassischen Software-Entwicklungszyklus entliehen sind. Das Vorgehen kombiniert das *top-down*- und *bottom-up*-Vorgehen zu einem hybriden Vorgehensmodell, was zur Entwicklung von Geschäfts- und Systemrollen führt, was sich insbesondere in den ersten beiden Phasen zeigt.

In der Analysephase steht im Vordergrund, den aktuellen Datenbestand zu erfassen und die Anforderungen des Kunden an den Einsatz von BRBAC zu formalisieren. Dazu wird die direkte Kommunikation und Interviews als Methodik verwendet. Die Erfassung des Ist-Zustands unterteilt sich in zwei voneinander unterschiedliche Bereiche, den Geschäfts- und den Systembereich. Es entsteht somit ein Lastenheft, das neben den Anforderungen des Kunden geschäftliche und technische Aspekte auf geeignetem Abstraktionsniveau formal erfasst. Die Verifikation dieser Artefakte durch den Kunden beschließt die Analysephase.

In der Entwurfsphase werden diese Artefakte dann spezifiziert und dabei verfeinert. Aus dem geschäftsnahen Ist-Zustand werden unter Beachtung der formalisierten Kundenanforderungen per *top-down*-Vorgehen Geschäftsrollen entwickelt, die über Verantwortliche und Abteilungen verfügen. Auf technischer Ebene hingegen entstehen per *bottom-up*-Vorgehen Systemrollen als Zusammenfassung unterschiedlicher Policys. Das Lastenheft wird insgesamt zu einem Pflichtenheft konkretisiert, in dem die Ergebnisse der Rollenspezifikation Einzug erhalten. Auch diese Phase wird durch die Abnahme durch den Kunden beendet.

Die Implementierungsphase selbst besteht aus zwei Teilen; der Installation des Rollenmanagementwerkzeugs im Unternehmen zusammen mit einer Pilotphase für ausgewählte Benutzer einerseits und dem Testbetrieb im Unternehmen und der Rückführung der daraus gewonnenen Ergebnisse ins Rollenmodell andererseits. Am Ende dieser Phase wird der Testbetrieb auf das gesamte Unternehmen ausgeweitet und abschließend durch den Kunden abgenommen, was anhand der Leistungsindikatoren und dem Pflichtenheft geschieht.

Die vierte Phase des Rollenbetriebs ist in das Vorgehen explizit eingeflossen, um zu signalisieren, dass es sich bei der Implementierung eines Rollenmodells um einen Lebenszyklus handelt. BRBAC muss fortwährend gewartet werden, um das Unternehmen als Ganzes stets in konsistenter Weise zu repräsentieren. Diese Phase ist daher eher als zyklisch, denn als iterativ zu bezeichnen und endet erst mit der Deprovisionierung der rollenbasierten Zugriffskontrolle, oder einer geplanten Änderung, was eine Rückkopplung in eine der davor liegenden Phasen nach sich zieht.

Dies beschließt die Entwicklung eines Vorgehensmodells zur Überführung von BRBAC in den Wirkbetrieb. Nachdem in den Kapiteln 3.2 und 3.3 aktuelle Rollenmanagementwerkzeuge bereits detailliert eingeführt wurden, folgt im nächsten Kapitel eine Bewertung dieser beiden Werkzeuge. Dazu wird zunächst ein Kriterienkatalog entwickelt, an dem die Werkzeuge bewerten werden. Diese Tätigkeit steht somit in Bezug zur Entwurfsphase, in der die Auswahl eines geeigneten Werkzeugs vorgenommen wird. Im weiteren Verlauf dieser Arbeit wird dann eines dieser Werkzeuge ausgewählt und darin die in den Kapitel 4 und 5 entwickelten Modelle zur

Ausführung gebracht. Die Bewertung sowie die Entwicklung eines Bewertungskatalogs sind Bestandteil des folgenden Kapitels.

6 ANALYSE AKTUELLER ROLLENMANAGEMENT-WERKZEUGE

Nachdem in Kapitel 4 mit BRBAC ein Rollenmodell für die rollenbasierte Zugriffskontrolle vorgestellt wurde, die den Anforderungen aus der Geschäftswelt sowie der Wissenschaft gleichermaßen gerecht werden soll und in Kapitel 5 ein Vorgehensmodell definiert wurde, um BRBAC im Wirkbetrieb erfolgreich umzusetzen, wird in diesem Kapitel nun eine Bewertung aktueller kommerzieller Werkzeuge aus dem Bereich des Rollenmanagements vorgenommen. Im Bezug auf das Vorgehensmodell findet diese Tätigkeit (engl. *product selection*) während der Analysephase statt, mit dem Ziel, ein Werkzeug auszuwählen, was im gegebenen Kontext am Besten geeignet ist. Dies wird in Information 47 grafisch verdeutlicht. Zur Produktauswahl wird nun zunächst ein Kriterienkatalog entwickelt und dieser im weiteren Verlauf des Kapitels auf zwei kommerzielle Werkzeuge angewandt. In einem Industrieprojekt, welches BRBAC auf der Basis des Vorgehensmodells in einem Unternehmen umsetzt, kämen mit Sicherheit mehr als zwei Werkzeuge in Betracht, jedoch würde dies den Rahmen dieser Arbeit sprengen. Dieser Kriterienkatalog kann aber zur Evaluierung weiterer Werkzeuge herangezogen werden. Auf der Grundlage der Kriterien, die im Katalog spezifiziert werden, wird insbesondere die Mächtigkeit des Rollenmodells gemessen, das in den zwei zur Auswahl stehenden Werkzeugen implementiert ist. Beide Werkzeuge, zusammen mit ihren Software-Architekturen und Komponenten sind in den Kapiteln 3.2 und 3.3 bereits angesprochen und wertungsfrei vorgestellt worden. Um eine Vergleichbarkeit dieser teils sehr unterschiedlichen Werkzeuge zu ermöglichen, wird die Metrik der jeweiligen Kriterien in Kapitel 6.1 so gewählt, dass sie eine Vergleichbarkeit unterschiedlicher Werkzeuge im Allgemeinen ermöglichen. Aus diesem Grund kann der Kriterienkatalog in weiteren Forschungsarbeiten auch auf andere Werkzeuge angewandt werden. Nach der Entwicklung des Katalogs folgt in den Kapiteln 6.2 und 6.3 dann schließlich dessen Anwendung auf die beiden Werkzeuge.

6.1 Entwicklung eines Kriterienkatalog

6.1.1 Motivation des Katalogs

In diesem Unterkapitel soll ein Kriterienkatalog entwickelt werden, mit dessen Hilfe Rollenmanagementwerkzeuge bewertet werden können. Die Kriterien orientieren sich dabei an typischen Aufgabenbereichen von Rollenmanagement im Wirkbetrieb und sind möglichst disjunkt zueinander gewählt, um eine breite Bewertungsgrundlage zu schaffen. Diese typischen Aufgabenbereiche wurden bereits in Kapitel 2.3 vorgestellt. Da die Kriterien unterschiedliche Werkzeuge und deren Architekturen vergleichen sollen, kann eine einheitliche Metrik für die gewählten Kriterien nicht gegeben werden. Stattdessen werden die Werkzeuge für jedes Kriterium individuell eingestuft und miteinander in Bezug gesetzt. Es wurde darauf geachtet, die typischen Funktionen einer Rollenmanagementanwendung ganzheitlich zu erfassen, wie etwa die Identifikation von Rollen – als initialer Ausgangspunkt beim Einsatz einer derartigen Lösung – oder die Möglichkeiten zur Implementierung des Rollenmodells sowie die Wartung der Lösung im produktiven Einsatz. Insgesamt betrachtet sind die unterschiedlichen Kriterien so gewählt, dass hieraus ersichtlich wird, ob ein Werkzeug eine angemessene Unterstützung beim Einsatz von BRBAC anbietet, oder nicht. Nachdem der Aufbau des Kriterienkatalogs und dessen Bedeutung für diese Arbeit erläutert wurden, folgt nun eine detaillierte Vorstellung der einzelnen Kriterien.

Das erste Kriterium befasst sich mit der Konzeption von Rollen in einem Rollenmanagementwerkzeug. Damit ist gemeint, was ein derartiges Werkzeug unter einer Rolle versteht und welche Granularität die implementierten Rollen besitzen. Es erfasst, zu welchem Grad Rollen als allgemeine Einheiten zur Kapselung von Benutzern und Policys unterstützt werden. Im Kontext dieser Arbeit ist es elementar, dass ein entsprechendes Werkzeug Rollen in differenzierter Form anbietet. Es sollte insbesondere zwischen geschäftsprozessnahen Geschäftsrollen und endsystemnahen Systemrollen unterscheiden können. Das zweite Kriterium setzt sich mit der Mög-

lichkeit auseinander, Hierarchien aufzubauen. Hierarchien sollten unterstützt werden, weil dadurch die Unternehmensstrukturen in angemessener Weise abgebildet und Berechtigungen strukturiert werden können. Durch die Möglichkeit der Vererbung kann so innerhalb von Geschäftshierarchien die Anzahl der Rollen auf einem überschaubaren Niveau gehalten werden. Dies ist wichtig, weil rollenbasierte Zugriffskontrollarchitekturen einerseits eine konsistente Struktur ermöglichen und andererseits die Komplexität bei der Verwaltung verringern möchten. Eine zweite Komplexität, die durch den Einsatz von RBAC erst entsteht, ist der zeitliche und personelle Aufwand, eine rollenbasierte Zugriffskontrollarchitektur zu implementieren. Das dritte Kriterium befasst sich daher mit dieser Komplexität und bewertet, ob eine Automatisierung beim Entwickeln von Rollen angeboten wird. Ein weiteres Kriterium ist die Unterstützung von Arbeitsprozessen bei der Wartung des Rollenmanagementwerkzeugs, weil sich Rollen im Laufe der Zeit entwickeln und sowohl die Rollen, als auch die Rollenstruktur angepasst werden müssen. Dieses Kriterium richtet sich demnach an die Akzeptanz des Werkzeugs. Beispielsweise ändern sich Zugriffsrechte, neue Rollen kommen hinzu oder die Mitgliedschaft von Benutzern in Rollen ändert sich. Information 51 zeigt den Bewertungskatalog, an dem die Werkzeuge gemessen werden. Das fünfte Kriterium versucht, eine Bewertung der Benutzerfreundlichkeit zu geben, obwohl dies objektiv nicht gegeben werden kann, weil dies stets von subjektiven Eindrücken beeinflusst wird. Jedoch wird hierbei versucht, den nötigen Arbeitsaufwand zu vergleichen, der nötig ist, um mit dem Werkzeug zu interagieren, was durchaus objektiv festgestellt werden kann. Das sechste Kriterium befasst sich mit der Art und Weise, wie die Daten des Rollenmanagementwerkzeugs in die Endsysteme zurückgeführt werden und somit mit den Kommunikationsbeziehungen zwischen den Systemen. Im Rahmen dieser Arbeit ist es eine grundlegende Voraussetzung, dass die rollenbasierte Zugriffskontrolle nicht für ein einzelnes Endsystem, sondern systemübergreifend eingesetzt wird. Aus diesem Grund macht es Sinn, die Kommunikationsbeziehung zu technischen Endsystemen zu analysieren. Beim Vorhaben, eine Zugriffskontrollarchitektur in eine bestehende Umgebung zu integrieren, die für diese autoritativ sein soll, stellt sich die Frage, ob sie im Sinne eines Dienstgeber/Dienstnehmer-Verhältnisses auch über Webservice-Schnittstellen erreichbar ist, oder nicht. Heutzutage ist eine Tendenz hin zu dienstorientierten Zugriffskontrollarchitekturen erkennbar [Em08, DM08], so dass das letzte Kriterium bewertet, ob die Rollenmanagementwerkzeuge diesem Paradigma folgen.

Criteria

1. Roles	
1.1 Business Roles	
1.2 System Roles	
2. Hierarchies	
2.1 Business Role Hierarchies	
2.2 System Role Hierarchies	
3. Role Mining	
4. Workflow capability	
4.1 Process	
4.2 Monitoring	
5. Usability	
6. Support of Feedback	
7. Service-oriented approach	

Information 51: Analysis of Role Management Solutions – Evaluation Criteria

6.1.2 Spezifikation der Kriterien

Die in Information 51 aufgeführten Kriterien werden im Folgenden detailliert beschrieben. Wie bereits angesprochen wurde, sind die genannten Kriterien disjunkt zueinander. Daher ist es nicht möglich, eine Metrik zu definieren, die für jedes Kriterium sowohl gültig, als auch aussagekräftig ist. Aus diesem Grund wird jedes Kriterium für sich in zwei Qualitätsausprägungen eingeteilt und innerhalb dieser in zwei Güteklassen. Die erste Unterscheidung gibt dabei an, ob das Werkzeug das Kriterium überhaupt erfüllt oder nicht. Dies wird symbolisiert durch ein Plus- bzw. ein Minussymbol. Wird eine Unterstützung eines Kriteriums durch das Werkzeug angeboten, wird dann die Qualität der Unterstützung dieses Kriteriums durch ein weiteres Plus- bzw. Minuszeichen verdeutlicht. Diese Werkzeugbeurteilungen wird dabei unabhängig vom jeweils anderen Werkzeug vorgenommen, um so den Bezug zur Idealanforderung der Kriterien herzustellen. Es entstehen demnach drei mögliche Beurteilungen für ein Kriterium, von zwei Plussymbolen für eine ausgezeichnete Unterstützung des Kriteriums, bis zu einem Minussymbol für keinerlei Unterstützung hierfür.

- **Kriterium 1: Rollen**. Wie bereits angesprochen wurde, haben Mitarbeiter in unterschiedlichen Abteilungen innerhalb eines Unternehmens unterschiedliche Sichtweisen auf die Gesamtstruktur des Unternehmens, geprägt durch deren technisches oder kaufmännisches Verständnis. Um zu einer erfolgreichen Umsetzung von RBAC zu gelangen, muss das System an diese Sichtweisen angepasst werden können. Dazu ist es nötig, dass sie auf geeignetem Abstraktionsniveau kombiniert werden können. Dies manifestiert sich in einer differenzierten Auffassung des Rollenbegriffs. Daher stellt die Unterscheidung zwischen geschäftsprozessnahen Geschäftsrollen und technikbezogenen Systemrollen einen sehr wichtigen Aspekt dar. Das erste Bewertungskriterium steht deshalb dafür, ob bzw. in welchem Maße die vorliegenden Werkzeuge diesen Sachverhalt unterstützen.

- **Kriterium 2: Hierarchien.** In den meisten Unternehmen existieren mehrere voneinander unabhängige Hierarchien, die sich nur schwer formal korrekt und konsistent erfassen lassen. Auf Geschäftsebene stellt sich dieser Sachverhalt folgendermaßen dar: Hier existieren Abteilungen, die aufeinander aufbauen und klar voneinander abgegrenzt sind. Im Allgemeinen existieren auch innerhalb von Abteilungen eine oder mehrere klare Hierarchien. Somit kann die Geschäftshierarchie klar definiert werden. Anders verhält es sich allerdings, wenn man statt der Strukturen nun die Berechtigungen auf geschäftlicher und technischer Ebene betrachtet: Hier sind individuell sehr unterschiedliche Rechte vergeben, was den Einsatz von RBAC deutlich erschwert. Im Gegensatz dazu lassen sich Rechte auf Geschäftsebene oft nicht formal erfasst, sondern sind lediglich implizit vorhanden. Ein Hauptziel einer rollenbasierten Struktur ist es, die Komplexität des Gesamtsystems zu verringern. Dies führt dazu, dass der Funktionsumfang von Rollen so konzipiert wird, dass diese möglichst effizient wiederverwendet werden können. Da die Aufgabenfelder der Abteilungen aber sehr unterschiedlich sind, ist es nicht trivial, möglichst generische Rollen zu konzipieren, die unterschiedlichen Kontexten gleichermaßen gerecht werden. Die Ausprägung einer Hierarchie, in der Berechtigungen vererbt werden, fällt somit sehr schwer. Das zweite Kriterium bewertet, ob die eingesetzten Werkzeuge eine Hierarchie auf Geschäftsebene sowie auf technischer Ebene unterstützen. Dazu zählt auch, in welchem Maße Berechtigungsvererbung ermöglicht wird.

- **Kriterium 3: Automatisiertes Entwickeln von Rollen.** Es ist eine realistische Annahme, davon auszugehen, dass Unternehmen über eine gewachsene Struktur mit verteilten Systemen verfügen [Ka07c]. Bei der Integration eines Rollenmanagementwerkzeugs wird somit initial ein gewisser Aufwand verursacht, der von der Migration der existierenden Unternehmensdaten herrührt. Der Erfolg von Rollenmanagementwerkzeugen hängt somit maßgeblich davon ab, ob diese das Herausarbeiten von Rollen aus einer bestehenden Unternehmensstruktur unterstützen. Das Anwenden von Algorithmen

zur analytischen und automatisierten Entwicklung allgemeiner Rollen (engl. *role mining*) stellt mittlerweile einen bewährten Ansatz in diese Richtung dar. Dieses Kriterium bewertet nun, ob das Werkzeug Algorithmen hierfür bereitstellt.

- **Kriterium 4: Unterstützung von Arbeitsabläufen.** Da an den Prozessen zum Aufbau einer rollenbasierten Struktur unterschiedliche Abteilungen beteiligt sind, ist es wichtig, diese Abläufe im eingesetzten Werkzeug zunächst abbilden und anschließend überwachen zu können. Die Abbildung gewährleistet eine zielgerichtete Durchführung der Abläufe im Querschnitt über alle Beteiligten, wohingegen die Überwachung sicherstellt, dass die an der Umsetzung beteiligten Personen die Fristen und Ziele bei den unterschiedlichen Abläufen einhalten. Das Werkzeug muss demnach prozessorientierte Arbeitsabläufe ermöglichen. Das vierte Kriterium bewertet die Unterstützung des Umsetzungsprozesses und deren Qualität.

- **Kriterium 5: Benutzerfreundlichkeit.** Das fünfte Kriterium bewertet die Benutzerfreundlichkeit des Werkzeugs. Dabei soll dediziert nicht auf den Bereich von Benutzerfreundlichkeitsuntersuchungen eingegangen werden, sondern lediglich eine subjektive Einschätzung der Benutzerführung der Werkzeuge gegeben werden. Das Hauptziel einer rollenbasierten Architektur ist es, eine Struktur zu schaffen, die die Komplexität des Gesamtsystems verringert und so einerseits Kosten spart, aber auch für mehr Konsistenz in der Umgebung sorgt. Ein deshalb nicht zu unterschätzender Faktor bei der Marktdurchdringung solcher Werkzeuge stellt die Bedienbarkeit dar. Da dies ein sehr individueller und subjektiver Aspekt ist, kann die Metrik dieses Kriteriums nicht objektiv gewählt werden, stützt sich aber auf Erfahrungswerte, die sich bei der Umsetzung des IST-Szenarios ergeben haben, welches beiden Werkzeuguntersuchungen gleichsam zugrunde liegt. Im Hinblick auf die Akzeptanz einer Rollenmanagementanwendung sind deren klare Strukturierung und eingängige Bedienbarkeit deshalb unerlässliche Merkmale.

- **Kriterium 6: Unterstützung von Rückmeldungen.** Zum jetzigen Zeitpunkt werden Rollenmanagementwerkzeuge in erster Linie als Verwaltungsplattform verwendet. Die Kommunikation zu den technischen Endsystemen stellt einen wichtigen Aspekt bei der Implementierung eines Rollenmanagementwerkzeugs dar. Die zu erteilenden Rollenmitgliedschaften und Berechtigungen werden von den Verantwortlichen zum Teil manuell umgesetzt. Jede manuelle Aktion birgt aber die Gefahr, entweder überhaupt nicht, oder nicht korrekt umgesetzt zu werden; das führt aber zu Inkonsistenzen zwischen der Verwaltungsplattform und den Endsystemen. Somit stellt sich neben der Frage nach der Kommunikationsbeziehung überdies die Frage, ob damit umgegangen werden kann, dass die erteilten Berechtigungen in den dahinterliegenden Endsystemen nicht mit den erteilten Berechtigungen übereinstimmen, die im Verzeichnis des Rollenmanagementwerkzeugs vorgehalten werden.

- **Kriterium 7: Serviceorientierter Ansatz**. Da Rollenmanagementwerkzeuge als integrative Lösung für den Einsatz in Unternehmen bestimmt sind, bewertet das siebte Kriterium, ob es eine Unterstützung für serviceorientierte Architekturen im Bereich des Identitätsmanagements bereitstellt. „Identity as a service“ verkörpert das Architekturprinzip der Serviceorientierung übertragen auf den Bereich des Identitätsmanagements [Em08]. Es bietet die Möglichkeit einer einfacheren Integration und Interaktion von identitätsbezogenen Anwendungen im Vergleich zu traditionellen Software-Architekturen.

6.2 Bewertung des Omada Identity Manager

6.2.1 Verknüpfung zum Stand der Technik

Nachdem der Omada Identity Manager (OIM) in Kapitel 3.2 bereits detailliert eingeführt und betrachtet wurde, folgt nun eine Bewertung dieses Werkzeugs anhand des soeben vorgestellten Kriterienkatalogs, mit dem Ziel, die Güte dieses Werkzeugs festzustellen. Das Ergebnis der Bewertung ist dargestellt in Information 52. Die Kriterien in dieser Abbildung stehen dabei im unmittelbaren Bezug zu den Unterkapiteln von „Rollenbasierte Zugriffskontrolle im Omada Identity Manager“ aus Kapitel 3.2. Im Folgenden soll kurz der Bezug zu diesen Kapiteln hergestellt werden. Die Bewertung des Kriteriums „Rollen“ (engl. *roles*) bezieht sich auf die Kapitel 3.2.1 sowie 3.2.3, in denen das in OIM verwendete Rollenmodell und die differenzierte Betrachtung von Geschäfts- und Systemrolle dargestellt wird. An dieser Stelle wird auch das Konzept der Rollenhierarchien erläutert, was in das Bewertungskriterium „Hierarchien“ (engl. *hierarchies*) einfließt. Da OIM in der vorliegenden Version keine Mechanismen zur automatischen Herausbildung von Rollen (engl. *role mining*) anbietet, steht dieses Kriterium im Katalog in keinem Zusammenhang zu einer der aufgeführten Unterkapitel. Die Unterstützung von automatischen Arbeitsabläufen (engl. *workflows*) kommt insbesondere in den Teilkapiteln 3.2.3 und 3.2.4 zum Tragen. Diese sind in OIM in den Arbeitsbereichen *role management* und *compliance* enthalten. Das Bewertungskriterium „Unterstützung von Rückmeldung“ (engl. *support of feedback*) bewertet, ob die in OIM gemachten Änderungen in automatischer oder automatisierbarer Form in die Zielsysteme zurückgeführt werden können und steht somit im Zusammenhang mit dem Arbeitsbereich *connectivity* aus Kapitel 3.2.2. Das Bewertungskriterium „Benutzerfreundlichkeit“ (engl. *usability*) ist querschnittlich über das gesamte Produkt zu verstehen Hier wird die Güte des gesamten Produkts bewertet, anstatt eines einzelnen Arbeitsbereichs oder einer einzelnen Komponente. Es ergibt sich daher nicht aus einem einzelnen Teilkapitel. Das letzte Kriterium „Serviceorientierter Ansatz“ (engl. *service-oriented approach*) bewertet, ob das Produkt im Sinne einer dienstorientierten Architektur auch als Dienstgeber fungieren kann, was elementar ist, um als zentrale Zugriffskontrollarchitektur angesehen werden zu können. Es muss somit auch als ganzheitliches Kriterium gesehen werden. Auch hier entstammt die Bewertung aus der gesamten Betrachtung vom OIM, als aus einem einzelnen Teilkapitel.

Criteria - Omada Identity Manager

	Support	Quality
1. Roles		
1.1 Business Roles	⊕	⊖
1.2 System Roles	⊕	⊖
2. Hierarchies		
2.1 Business Role Hierarchies	⊕	⊕
2.2 System Role Hierarchies	⊕	⊕
3. Role Mining	⊖	
4. Workflow capability		
4.1 Process	⊕	⊕
4.2 Monitoring	⊕	⊕
5. Usability	⊕	⊖
6. Support of Feedback	⊕	⊕
7. Service-oriented approach	⊕	⊖

Information 52: Analysis of Role Management Solutions – Evaluation of OIM

6.2.2 Anwendung des Kriterienkatalogs

In Information 52 ist die Bewertung des Omada Identity Manager aufgeführt. Die sieben Kriterien werden im Folgenden nun einzeln bewertet.

Rollen

Der Omada Identity Manager (OIM) bietet ein Rollenmodell, welches sehr individuell anpassbar und somit sehr flexibel ist. Dies führt jedoch auch dazu, dass – abweichend vom RBAC-Standard – Objekte als Rolle definierbar sind, die es der Definition nach nicht sind. So kann in OIM etwa ein Benutzer selbst eine Rolle darstellen. Die Unterscheidung unterschiedlicher Rollentypen geschieht in OIM über frei wählbare Rollenkategorien (engl. *role categories)*. Es existieren beispielsweise die Kategorien „Systemrollen" und „Jobprofile", wodurch Geschäfts- von Systemrollen unterschieden werden können. Rollenkategorien werden als Attribute von Rollen dargestellt, so dass die Unterscheidung zwar vorhanden ist, wenn auch nur implizit. Eine Erschwernis im Umgang mit OIM stellt ein weiteres Attribut von Rollen dar, das ebenfalls verwendet wird: Jede Rolle verfügt über einen „Rollentyp" (engl. *role type*), wodurch die Unterscheidung von Geschäfts- und Systemrollen redundant vorgenommen wird. An dieser Stelle wird von OIM keine Konsistenzüberprüfung vorgenommen, so dass eine Rolle vom Typ „Geschäftsrolle" auch gleichzeitig in die Rollenkategorie „Systemrolle" eingeteilt werden kann. Der Identity Manager ermöglicht es im Rahmen von Rollen, dass ein Benutzer über sogenannte „sekundäre Identitäten" (engl. *secondary identities*) verfügt, um es einem Benutzer zu ermöglichen, über das durch seine Rollenmitgliedschaften definierte Maß hinaus zusätzliche Berechtigungen zu erhalten. Dies widerspricht allerdings dem RBAC-Paradigma, wonach die Berechtigungen eines Benutzers an dessen Rollen und nicht direkt an dessen Identität geknüpft werden. Zur Definition von wechselseitigem Ausschluss (engl. *separation of duty*, SoD) bietet OIM *constraints* auf zentraler Ebene an. Dabei ist es möglich, eine Menge an Rollen auszuwählen, die sich wechselseitig ausschließen.

Hierarchien

In OIM wird eine Vielzahl unterschiedlicher Hierarchien ermöglicht: Zunächst existiert eine Hierarchie auf der Ebene von Rollen, so dass sich Hierarchien bei Geschäfts- und Systemrollen definieren lassen. Durch die fehlende Unterscheidung dieser beider Rollentypen vermischen sich allerdings geschäftsnahe und technikspezifische Hierarchien. Eine davon unabhängige Hierarchie ordnet die Rollen in Organisationseinheiten (engl. *organizational unit*) an, die frei wählbar sind. Hierdurch werden die Strukturen der Organisation wie etwa Abteilungen abgebildet. Eine Rolle ist dabei stets an genau eine Organisationseinheit geknüpft, was zwar die Rollen verallgemeinert, die alle Benutzer in dieser Organisationseinheit besitzen, aber eine Wiederverwendung dieser Rollen in anderen Abteilungen ausschließt. Eine weitere hierarchische Struktur ist durch die sogenannten „Rollenordner" (engl. *role folders*) gegeben, die eine Einteilung der Rollen in Geschäftsrollen und Systemrollen erlaubt, wobei die Systemrollen selbst nach der Zugehörigkeit zu technischen Endsystemen unterteilt sind. Auch an dieser Stelle zeigt sich die Flexibilität von OIM, weil hierbei eigene Rollenordner angelegt werden können, die eine logische Gruppierung unterschiedlicher Rollentypen zulässt. Eine Wahrung der Konsistenz ist auch hierbei nicht gegeben, da eine Rolle ungeachtet des Typs oder der Kategorie in Ordner eingeteilt werden kann. Auch dieses Attribut lässt lediglich eine eindeutige Zuordnung zu Rollenordnern zu, was eine mehrfache Verwendung der definierten Rolle verhindert. OIM ermöglicht durch die eingebauten *workflows*, dass die Rollenzuweisungen und somit auch die darin definierten Rechte in Abhängigkeit der Zugehörigkeit zu Organisationseinheiten automatisch vergeben werden. So werden etwa die effektiven Rollen eines Benutzers automatisch aktualisiert, sobald dieser Benutzer in eine andere Organisationseinheit eingeteilt wird. Von dieser automatischen Einteilung profitieren insbesondere Systemrollen, da diese zentral für eine Organisationseinheit definiert werden können und dadurch die Policy-Konformität für alle Benutzer dieser Organisationseinheit sicherstellen.

Algorithmische Entwicklung von Rollen

In der vorliegenden Version bietet OIM keine Unterstützung für das algorithmische Entwickeln von Rollen (engl. *role mining*) an. Dies führt dazu, dass der initiale Aufwand sehr hoch ist, der nötig ist, um eine gewachsene Struktur auf eine rollenbasierte Zugriffskontrollarchitektur umzustellen. Es kann weder vorab analysiert werden, welche technischen Berechtigungen im Unternehmen vorhanden sind, um diese zu Systemrollen zusammenzufassen, noch existiert eine algorithmische Unterstützung für das Entwickeln von Geschäftsrollen. Dadurch, dass OIM auf dem Identity Lifecycle Manager (ILM) basiert, einem Provisionierungswerkzeug der Firma Microsoft, können allerdings Daten aus dem Gesamtdatenbestand der Einrichtung als Basis der Entwicklung bezogen werden. Sollten die hierüber bezogenen Daten bereits Rollen aufweisen, können diese als Grundlage des Rollenmodells in OIM verwendet werden.

Automatische Arbeitsabläufe

OIM bietet neben einer eigenen Ausführungseinheit für automatische Arbeitsabläufe (engl. *workflow engine*) auch eine in die grafische Benutzeroberfläche integrierte Entwurfskomponente (engl. *workflow designer*) an. Diese lässt eine sehr individuelle Anpassung von Geschäftsprozessen zu, wobei an dieser Stelle auch Verantwortlichkeiten im Prozess definiert werden können. Es werden auch bereits vorkonfigurierte Prozesse mitgeliefert, allerdings sind dies keine Geschäftsprozesse im eigentlichen Sinne, sondern automatische Abläufe für den Wirkbetrieb des Rollenmanagementwerkzeugs selbst. So wird etwa bei der Einteilung eines Benutzers in eine Organisationseinheit in Abhängigkeit von der Geschäftsrolle und SoD-*constraints* die Menge an Rollen gesammelt und in der Summe automatisch an den Benutzer verteilt. Für die Überprüfung von *workflows* oder etwa die Planung von Umstrukturierungen bietet OIM keinerlei Unterstützung an. Bei geplanten Änderungen kann nicht überprüft werden, wie sich diese auf das aktuelle Rollenmodell auswirken würden, weil die Datentypen in OIM aktiv sind, sobald sie definiert wurden. Dies trifft somit auch auf Beschränkungen wie etwa SoD-*constraints* zu, wodurch eine Planung somit auch hier nicht gewährleistet ist.

Benutzerfreundlichkeit

OIM bietet eine einheitliche, grafische Benutzeroberfläche an, die sehr intuitiv zu bedienen ist. Durch die Flexibilität des Rollenmodells ist es jedoch nötig, dass das Personal eingehend geschult wird. Ein zentrales Konzept bei OIM sind die rollenspezifischen Ansichten (engl. *views*), durch die die Interaktionsmöglichkeiten eines Benutzers in der Ausübung seiner Rolle mit OIM kanalisiert werden. Änderungen an diesen Ansichten sind jedoch sehr zeitintensiv, weil es dafür keine zentrale Arbeitsoberfläche gibt. OIM unterstützt sehr stark die dezentrale Administration, so dass ein Benutzer Änderungen an seinen Rollenzugehörigkeiten selbst initiieren kann. Dabei werden *workflows* aktiviert und diese zum darin definierten Verantwortlichen delegiert. Dies erhöht die Benutzerfreundlichkeit und somit potentiell die Akzeptanz des Werkzeugs im Wirkbetrieb. Zusätzlich dazu erhöht sich durch die dezentrale Administration die Effizienz im Unternehmen, weil Änderungen nicht mehr nur davon abhängig sind, wann sie von der technischen Administration initiiert und bearbeitet werden, sondern davon, wann diese bereits initiierte Änderungswünsche evaluiert. OIM bietet auch eine Integration seiner grafischen Benutzeroberfläche in das Intranet-Portal Office SharePoint Server (MOSS) der Firma Microsoft an, indem die Komponenten des OIM als Web-Module (engl. *web part*, [Mic08]) angeboten werden. Ein letzter Aspekt zur Bedienbarkeit ist der Aspekt der Sprachanpassung. OIM ist ein *multi-language*-Werkzeug, das in mehreren Sprachen betrieben werden kann. Dies unterstützt in vielen Fällen den intuitiven Umgang für Benutzer. An dieser Stelle soll lediglich eine Grobeinschätzung der Benutzerfreundlichkeit gegeben werden, da für eine intensiven Bewertung dedizierte Studien anzustellen sind. Insgesamt betrachtet wird die Benutzerfreundlichkeit von OIM durch die *workflows,* die anpassbaren Ansichten sowie die dezentrale Administration als gut bewertet.

Kommunikation zu den Endsystemen

OIM basiert auf dem Provisionierungswerkzeug ILM, welches Änderungen in die Endsysteme zurückführt und außerhalb von OIM gemachte Änderungen rückgängig macht und gegebenen-

falls verwirft. Dies geschieht über die Managementagenten, die für jedes Endsystem erstellt werden müssen. Die Erstellung geschieht direkt in der Benutzeroberfläche, indem die in den zugrundeliegenden Endsystemen zu synchronisierenden Attribute und Daten selektiert werden.

Serviceorientierung

OIM unterstützt das serviceorientierte Paradigma, allerdings nur in eingeschränktem Umfang. Die erwähnten Agenten, die die Schnittstelle zwischen OIM und dem Gesamtdatenbestand des Unternehmens angesehen werden, verfügen über eine Webservice-Schnittstelle, so dass die in OIM verarbeiteten Daten im Sinne eines Dienstgeber/Dienstnehmer-Verhältnisses von OIM als autoritativer Quelle bezogen werden können. Dies ist insbesondere bei der Integration von RBAC mit anderen Zugriffskontroll-Architekturen von Vorteil.

6.3 Bewertung des Sun Role Manager

6.3.1 Bezug der Kriterien zum Stand der Technik

In diesem Teilkapitel wird der Sun Role Manager (SRM) anhand des definierten Kriterienkatalogs eingeordnet und bewertet. Die Grundlage hierfür stellt die Auseinandersetzung mit diesem Produkt aus den Kapiteln 3.3.1 bis 3.3.5 sowie die daraus gewonnen Erkenntnisse dar. Auch hier soll zunächst ein Bezug hergestellt werden zwischen den einzelnen Bewertungskriterien und den Unterkapiteln von Kapitel 3.3, ehe die Bewertung vorgenommen wird. Die Erfahrungswerte bezüglich der Rollen und Hierarchien steht dabei im engen Zusammenhang zu den Kapiteln 3.3.2 und 3.3.3, in denen die Bereiche Rollenmanagement und die Persistenz der Daten in SRM genau betrachtet werden. Die Bewertung der *role mining*-Funktionalität bezieht sich auf den Bereich *role engineering* aus Kapitel 3.3.3, in dem prozesshaft die automatisierte Entwicklung von Rollen aufgezeigt wird. Die Kapitel 3.3.4 sowie 3.3.5 befassen sich eher mit der Wartung des Rollenmodells im Wirkbetrieb und damit einhergehend mit dem gesamten Lebenszyklus von Rollen. Diese Erkenntnisse fließen in das Kriterium *workflow capability* ein. In Analogie zur Bewertung des Omada Identity Manager sind die abschließenden drei Kriterien querschnittlich über das gesamte Werkzeug zu verstehen. Die Benutzerfreundlichkeit ist dabei wiederum subjektiv geprägt, da jedoch beiden Werkzeugumsetzungen dasselbe Szenario zugrunde liegen, das in Kapitel 1.3 kurz vorgestellt wurde, kann der relative Vergleich zwischen beiden Werkzeugen durchaus als repräsentativ angesehen werden. Das Ziel dieser Aufteilung ist es auch hier, einen quantitativen Einblick in dieses Werkzeug zu geben.

Criteria - Sun Role Manager

	Support	Quality
1. Roles		
1.1 Business Roles	⊕	⊕
1.2 System Roles	⊕	⊖
2. Hierarchies		
2.1 Business Role Hierarchies	⊕	⊖
2.2 System Role Hierarchies	⊖	
3. Role Mining	⊕	⊖
4. Workflow capability		
4.1 Process	⊕	⊖
4.2 Monitoring	⊕	⊖
5. Usability	⊕	⊖
6. Support of Feedback	⊖	
7. Service-oriented approach	⊖	

Information 53: Analysis of Role Management Solutions – Evaluation of SRM

6.3.2 Anwendung des Kriterienkatalogs

Information 53 zeigt die Bewertung des Sun Role Manager am Kriterienkatalog. Im Folgenden wird nun einzeln auf die sieben Kriterien eingegangen und die Bewertung begründet.

Rollen

Der Sun Role Manager fasst eine Rolle generell als Geschäftsrollen auf, bietet aber auch eine native Unterstützung von Systemrollen an. Dazu werden Rollen mit einem Rollentyp versehen, der diesen Unterschied festlegt. Diese Festlegung ist allerdings lediglich eine semantische Spezifikation, die in SRM in der vorliegenden Version nicht beachtet wird. Mit den Policys wird ein Mechanismus angeboten, um technische Berechtigungen mit Geschäftsrollen direkt zu verknüpfen. Rollen verfügen neben einer Menge an Policys und einen Rollenverantwortlichen sowie Organisationseinheiten, in denen sie verfügbar ist insbesondere über eine Menge an Benutzern, die mit ihr verknüpft sind. Somit kann eine Geschäftsrolle mit den darin definierten Policys in unterschiedlichen Kontexten wiederverwendet werden. Benutzer in unterschiedlichen Abteilungen erhalten somit durch die Einteilung in eine Rolle alle der darin definierten Berechtigungen in den technischen Endsystemen. Geschäftsrollen werden versioniert, so dass Änderungen zurückverfolgt und wiederhergestellt werden können. In SRM wird der wechselseitige Ausschluss von Rollen im Gegensatz zu OIM nicht zentral vorgehalten, sondern in den betreffenden Rollen selbst festgehalten. Somit kann in einer Rolle direkt eingesehen werden, mit welchen anderen Rollen sie im Konflikt steht. Eine Rolle kann in SRM explizit aktiviert und deaktiviert werden sowie ein zeitliches Intervall angegeben werden, für das sie aktiv ist. Daher ist es möglich, zeitliche Beschränkungen zu definieren, die automatisch umgesetzt werden.

Hierarchien

SRM unterstützt eine Hierarchie bei Geschäftsrollen und darüber hinaus beliebig viele, voneinander unabhängige Hierarchien bei den Geschäftseinheiten, zu denen die Rollen zugeteilt sind. Leider beschränkt sich die Hierarchiebildung bei Rollen lediglich auf organisatorische Aspekte, wie etwa strukturelle Hierarchien auf Geschäftsebene, so dass gemeinsame Rechte, oder Eigenschaften im Allgemeinen nicht vererbt werden können. Eine Hierarchie auf Ebene der Policys wird daher in der vorliegenden Version nicht unterstützt. Daher können gemeinsame Rechte

nicht modularisiert werden und müssen in allen über dieses Recht verfügende Rollen eingetragen werden. Dies sorgt für einen gewissen Pflegeaufwand innerhalb des Systems und verursacht darüber hinaus mehrfache Datenhaltung. Daraus ergeben sich Mehraufwände in den Bereichen Datenkonsistenz und Datenadministration, was gerade in größeren Einrichtungen möglichst zu vermeiden ist. Da im Gegensatz zu Geschäftsrollen bei Systemrollen keine Hierarchie unterstützt wird, ist es nicht möglich ist, eine Vererbungsstruktur bei Systemzugriffsberechtigungen zu etablieren. Das *identity warehouse* als zentraler Datenspeicher ermöglicht es nicht, Geschäftsrollenattribute im Sinne von Referenzen auf miteinander zu verbinden, weil die Attributfelder reine Textfelder anstatt Verweisen sind. Somit muss die Hierarchie auf Ebene der Geschäftsrollen manuell administriert werden, was wiederum sowohl zu Inkonsistenzen, aber auch zu höherem Wartungsaufwand führt.

Algorithmische Entwicklung von Rollen

SRM bietet eine Unterstützung für den Bereich *role engineering* an und hierbei insbesondere Algorithmen zur automatisierten Entwicklung von Rollen (engl. *role mining*) an. Die *role mining*-Funktionalität kann durch Parameter angepasst werden, jedoch sind diese hierfür weder dokumentiert, noch intuitiv verständlich, so dass hierfür Detailwissen aus dem *data mining* vorhanden sein muss. Die Benutzerobjekte, auf denen die *role mining*-Algorithmen arbeiten, sind an maximal eine Geschäftseinheit gebunden. Dies ist für den arbeitsteiligen Aufbau gerade von großen Unternehmen sehr unflexibel, in der Rollen gleichzeitig in mehreren Abteilungen beschäftigt sind. Auch hier macht sich die fehlende Hierarchie bemerkbar: So ist eine Hierarchie für ein Unternehmen zwar möglich, aber die Benutzer sind immer genau einer Geschäftseinheit zugeordnet und nur in dieser sichtbar. Auf der Ebene der nächsthöheren Geschäftseinheit kann damit nicht auf einen Blick eingesehen werden, welche Benutzer in den tiefer liegenden Einheiten vorhanden sind, was es deutlich erschwert, einen Gesamtüberblick zu gewinnen.

Automatische Arbeitsabläufe

SRM ermöglicht durch die eingebaute *workflow engine*, Arbeitsabläufe zu teilautomatisieren, wodurch gerade im Fall verteilter Abteilungen eine Verbesserung bei Koordinierungsaufgaben zu erkennen ist. Auch lassen sich die Arbeitsabläufe selbst automatisiert überwachen, was dazu führt, dass auf Fristüberschreitungen innerhalb des Prozesses regiert werden kann. Beispielsweise kann nach Ablauf einer Frist ein Verantwortlicher darüber in Kenntnis gesetzt werden. Diese Fristen lassen sich systemweit einmalig definieren, was bei Unternehmen mit unterschiedlichen Prozessen ebenfalls sehr starr ist. Die drei vordefinierten Arbeitsabläufe orientieren sich sehr stark am Rollenmanagement und lassen sich durch einen grafischen Editor erweitern. Neue Abläufe lassen sich hier allerdings nicht definieren. Die Überwachung der Rolleneinteilungen bzw. Zertifizierungen lassen sich teilautomatisieren, in dem sie zu einem beliebigen Zeitpunkt in der Zukunft gestartet werden. Diese automatischen Abläufe sind für Arbeiten gedacht, die in regelmäßigen Zeiträumen wiederkehren, aber leider bietet der SRM in der vorliegenden Version nicht die Möglichkeit, diese Abläufe regelmäßig laufen zu lassen, sondern nur einmalig.

Benutzerfreundlichkeit

Die Bedienbarkeit, die zwar sehr individuell und somit subjektiv ist, stellt trotzdem einen sehr wichtigen Aspekt eines Rollenmanagementwerkzeugs dar. Es lässt sich feststellen, dass die Einarbeitungszeit für den Role Manager sehr hoch ist. Einerseits liegt das an der Struktur der grafischen Bedienoberfläche, die nur auf der ersten Ebene sehr klar gegliedert ist. Hier wird klar unterschieden zwischen den Komponenten, die der SRM bereitstellt. Innerhalb dieser Komponenten ist die Aufteilung der Arbeiten nicht so übersichtlich gegliedert. Andererseits liegt das an den missverständlichen Begrifflichkeiten. So existieren etwa in den verschiedenen Aufgabenbereichen unterschiedliche Auffassungen des Begriffs „Policy“. Im Allgemeinen ist damit eine Systemrolle gemeint. Im Falle von Zertifizierungen beschreibt eine Policy hingegen eine Attributmenge, die als Muster für Zugriffsverletzungen dient, um so Regelverletzungen zu erkennen.

Kommunikation zu den Endsystemen

SRM ist in der Standardinstallation ein nicht-invasives Rollenmanagementwerkzeug. Das bedeutet, dass es keine direkte Integration mit den bestehenden Unternehmenssystemen gibt. Stattdessen müssen die Daten aus den Unternehmenssystemen in regelmäßigen Abständen in Form von vorher exportierten CSV-Dateien manuell in den Sun Role Manager importiert werden. Dabei gilt es zusätzlich zu beachten, dass die Namenskonvention der Attribute im *identity warehouse* mit der Konvention in der zu importierenden CSV-Datei übereinstimmen muss, die von dem technischen Endsystem vorgegeben wird. Durch diesen mauellen Zwischenschritt ist es für SRM nicht möglich, zu überprüfen, ob die im *identity warehouse* etablierten Rechte in den Rollen mit den tatsächlichen Rechten im Endsystem übereinstimmen. Dies ist ein Sachverhalt, der sehr ungünstig ist, weil durch den Einsatz eines Rollenmanagementwerkzeugs der Aufwand verringert und nicht auf andere Verwaltungsaufgaben umgeschichtet werden sollte, so dass der resultierende Gesamtaufwand auf Seiten der technischen Administration letztlich in etwa gleichhoch ist. Ein teilautomatisiertes Zurückführen der in SRM etablierten Rollen ist aktuell nur mit dem Sun Identity Manager möglich, was in dieser Arbeit aber ausgeklammert wurde.

Serviceorientierung

Das letzte Kriterium der Serviceorientierung wird vom Sun Role Manager nicht adressiert. Es ist aktuell nicht möglich, über eine Webservice-Schnittstelle mit dem System zu interagieren. Dadurch ist es über das Maß an manuellem Export etwa über CSV-Dateien nicht möglich, von Außen mit dem Rollenmanagementwerkzeug zu kommunizieren.

6.4 Resümee

In diesem Kapitel wurde ein Kriterienkatalog entwickelt, um Rollenmanagementwerkzeuge vergleichbar zu machen. Dieser Kriterienkatalog orientiert sich dabei an typischen Aufgaben von Rollenmanagementwerkzeugen und definiert Kriterien, die sich möglichst nicht überschneiden, um das gesamte Spektrum eines derartigen Werkzeugs qualitativ zu erfassen. Die beiden wichtigsten Punkte dabei waren die Unterstützung von Geschäfts- und Systemrollen einerseits und die Ausprägung von Hierarchien für diese Rollentypen andererseits. Es lässt sich feststellen, dass aktuelle Werkzeuge aus dem Rollenmanagement grundlegende Aufgaben realisieren, die bei der Umsetzung einer rollenbasierten Zugriffskontrolle benötigt werden. Die Automatisierung von Aufgaben in Form von *workflows* wirkt sich dabei im produktiven Einsatz ebenso positiv aus, wie die Sicherstellung, dass Policy-Vorgaben eingehalten werden. Die technischen Umsetzungen verursachen jedoch zum jetzigen Zeitpunkt eine sehr hohe Arbeitsbelastung im Wirkbetrieb, was auch an der fehlenden expliziten Unterscheidung von geschäftlichen und technischen Aspekten liegt.

Nach der Entwicklung des Kriterienkatalogs wurden die beiden Werkzeuge Omada Identity Manager (OIM) und Sun Role Manager (SRM) an ihm bewertet. Vergleicht man nun die Ergebnisse beider Werkzeuge, so kann festgehalten werden, dass OIM klare Vorteile bei Hierarchien und der Vererbung von Rechten besitzt. Beides unterstützt die Arbeitsprozesse in der Rollenverwaltung, da die Rechte teils automatisch neu berechnet werden, sobald sich an den Rolleneinteilungen eines Benutzers Änderungen ergeben. Auch die dezentrale Administration ist als Vorteil von OIM gegenüber SRM hervorzuheben, da dies die Effektivität einer rollenbasierten Zugriffskontrolle erhöht. Die grafische Formulierung von Arbeitsabläufen und die benutzerspezifischen Ansichten führen potentiell zu einer hohen Akzeptanz des Werkzeugs im produktiven Einsatz, was sich in SRM so nicht wiederfindet. SRM hingegen bietet eine sehr ausgeprägte Unterstützung für das automatische Entwickeln von Rollen und die Wartung des Rollenmanagementwerkzeugs im Wirkbetrieb. Bezugnehmend auf das Vorgehensmodell aus Kapitel 5 kommen diese Komponenten in allen vier Phasen zum Tragen und unterstützen somit den Gesamtprozess zur Umsetzung einer rollenbasierten Zugriffskontrolle sehr stark. Die automatische Rückführung in die Endsysteme ist in beiden Werkzeugen durch eine eigene Provisionierungslösung gegeben. Der OIM verwendet den Identity Lifecycle Manager der Firma Micro-

soft und auch für den SRM existiert die Möglichkeit der Integration mit dem Identity Manager der Firma Sun, was in dieser Arbeit aber nicht betrachtet wurde. Diese Funktionalität ist elementar bei der Integration von Rollenmanagementwerkzeugen in einer gewachsenen Unternehmensstruktur, weil der manuelle Datenaustausch zu den Endsystemen die Konsistenz der Unternehmensdaten nicht sicherstellen kann und darüber hinaus äußerst zeitaufwändig ist. Als letzter Aspekt lässt sich feststellen, dass beide analysierten Werkzeuge in den vorliegenden Versionen das Paradigma der dienstorientierten Architekturen nicht hinreichend unterstützen. Mit dem Rollenmodell BRBAC aus Kapitel 4 wird eine explizite Trennung von Geschäfts- und Systemrollen gefordert und Erweiterungen dazu modelliert, die auf dieser expliziten Trennung basieren. Die in BRBAC geforderte Trennung lässt sich in den analysierten Werkzeugen nicht nativ umsetzen, weil das Rollenmodell NIST-RBAC, auf welchem diese basieren, die Unterscheidung nur implizit in Form von Rollenattributen vornimmt.

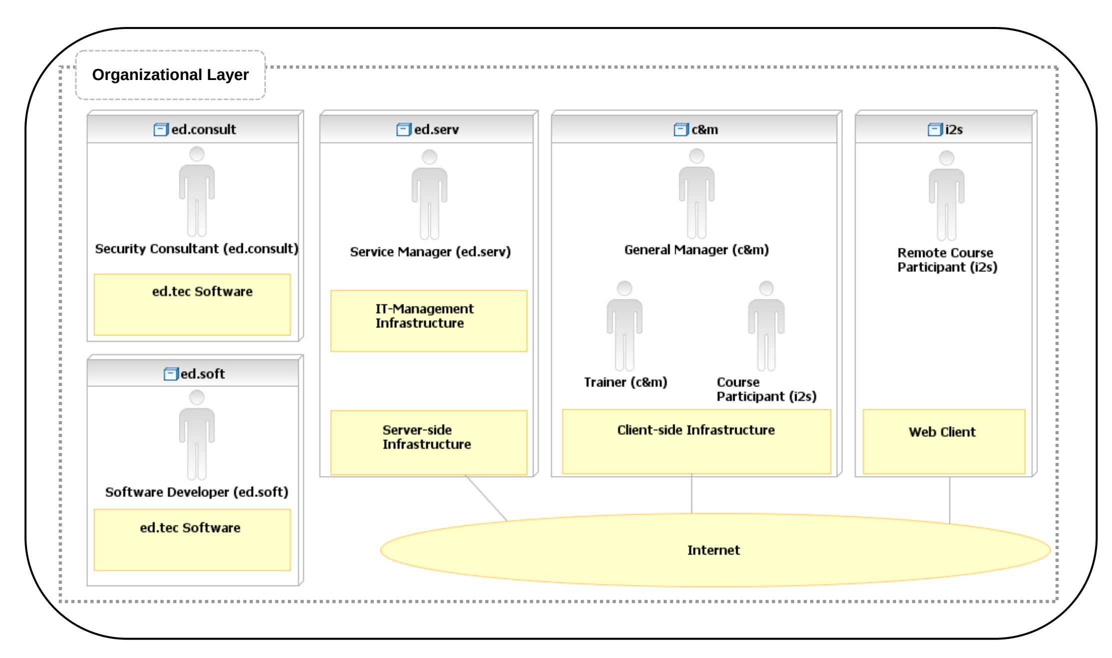
Im nächsten Kapitel wird auf der Basis der hier vorgenommenen Analyse der Werkzeuge dasjenige ausgewählt, das im vorgegebenen Szenario als vorteilhaft erachtet wird und BRBAC im Vorgehensmodell aus Kapitel 5 in den produktiven Einsatz zu überführen.

7 FALLSTUDIE ZUR ANWENDUNG DER MODELLE IN EINEM KOMMERZIELLEN PRODUKT

In diesem Kapitel soll die Tragfähigkeit der Rollenmodelle gezeigt werden, die in dieser Arbeit entwickelt wurden. Dazu wird zunächst ein Szenario gewählt, welches der Fallstudie zugrunde liegt. Dazu wird zunächst das Geschäftsmodell eines gedachten Unternehmens vorgestellt, welches auf eine rollenbasierte Zugriffskontrollarchitektur umsteigen möchte. Um dies zu realisieren, wird das Vorgehensmodell aus Kapitel 5 angewandt, welches das Rollenmodell aus Kapitel 4 in den Wirkbetrieb überführt. Anschließend wird auf der Basis der Werkzeugbewertung aus Kapitel 6 ein Rollenmanagementwerkzeug für dieses Szenario ausgewählt und Kernaspekte der technischen Umsetzung beleuchtet.

7.1 Vorstellung des Beispielszenarios *internet-supported training*

In diesem Kapitel wird das Beispielszenario *internet-supported training* (IST) vorgestellt, welches der Umsetzung der Modelle zugrunde liegt, die in dieser Arbeit entwickelt wurden. Dabei werden zunächst die an diesem Szenario beteiligten Unternehmen präsentiert und anschließend die Geschäfts- und Systemrollen definiert. Dieses Szenario handelt von einem Anbieter IT-gestützter Fortbildungskurse *cooperation&more* (c&m). Die Kernkompetenz des Unternehmens c&m stellt das Angebot von Fortbildungskursen über das Schulungsportal *education technology* (ed.tec) dar, welches von der Software-Firma *education software* (ed.soft) vertrieben wird. Ein exemplarischer Kunde von c&m ist das Unternehmen *intelligent internet solutions* (i2s), das seine Mitarbeiter durch das Schulungsangebot von c&m fortbilden lässt. Dabei kann die Teilnahme an Fortbildungskursen sowohl per Fernzugriff aus dem internen Netz von i2s erfolgen, aber auch per direktem Zugriff aus der technischen Infrastruktur von c&m. Für diesen Fall verfügen die Mitarbeiter von i2s über Benutzerkonten in der Umgebung von c&m, um sich an dort zur Verfügung gestellten Rechnerarbeitsplätzen anmelden zu können. Die IT-Infrastruktur von c&m ist dabei unterteilt in Client-Rechner, die von c&m selbst verwaltet werden und Serversysteme, die vom IT-Dienstleister *education services* (ed.serv) gepflegt werden. Da die Pflege der IT-Landschaft weder die Kernkompetenz von c&m, noch von ed.serv ist, ist c&m seinerseits Kunde des Consulting-Unternehmens *education consulting* (ed.consult), das c&m bei der Konsolidierung seiner Unternehmens-IT unterstützt. Information 54 zeigt eine Übersicht über die am IST-Szenario beteiligten Unternehmen.



Adapted from [C&M-A-BA], page 10

Information 54: Case Study – Introduction to IST-scenario

7.1.1 Vorstellung der Geschäftsrollen

In diesem Teilkapitel werden die Geschäftsrollen beschrieben, die im IST-Szenario vorkommen und die Beziehungen erläutert, die zwischen Ihnen existieren. Ausgehend von den Geschäftsrollen des Unternehmens c&m werden die Geschäftsrollen der anderen am IST-Szenario beteiligten Unternehmen vorgestellt. Information 54 zeigt bereits einige der Rollen auf, jedoch werden für diese Fallstudie zusätzliche Rollen definiert, die in der Abbildung der Übersichtlichkeit halber nicht aufgeführt wurden.

- **Der Schulungsanbieter c&m.** Das Unternehmen c&m wird vertreten durch die Unternehmensleitung, welche für die Koordination der unterschiedlichen Abteilungen von c&m zuständig ist. Dabei gibt es erstens die Abteilung „Marketing" mit der Geschäftsrolle „Marketing-Assistent", die für die Akquise neuer Kunden zuständig ist. Die zweite Abteilung „Buchhaltung" wird vertreten durch die Rolle „Buchhaltungsassistent" und kümmert sich um die Verträge mit den Kunden von c&m und die Rechnungen an die Partner von c&m. Eine dritte Kompetenz der Buchhaltung ist die Verwaltung sämtlicher c&m-interner Verträge und Zahlungen. Die dritte Abteilung „IT-Administration" kümmert sich um die Pflege der Server- und Client-Computersysteme, die sich in der Infrastruktur von c&m befinden. Die Administration verfügt über die Geschäftsrolle „System-Administrator" und stehen in direktem Kontakt zu den Firmen ed.serv sowie ed.consult. Eine vierte Abteilung von c&m kümmert sich um die Erzeugung und Pflege der Inhalte, die den Kunden von c&m zur Verfügung gestellt werden. Dafür existiert die Geschäftsrolle „Autor", die mit der unternehmenseigenen IT-Abteilung, den Kunden von c&m und der Marketing-Abteilung kommuniziert. Für die Kunden von c&m existiert eine eigene Geschäftsrolle „Kunde".

- **Der Schulungskunde i2s.** Das Unternehmen i2s besteht zum Einen aus der Unternehmensleitung, die in direktem Kontakt zur Unternehmensleitung von c&m steht. Auch die Marketing-Abteilung von c&m steht mit der Unternehmensleitung von i2s in direktem Kontakt, um über neue Kursangebote zu informieren. Zum Anderen verfügt das Unternehmen i2s über eine Menge an Kursteilnehmern, die in der Geschäftsrolle „Kursteilnehmer" an Schulungen von c&m teilnehmen. Die Teilnahme kann, wie bereits erwähnt wurde, sowohl aus der technischen Infrastruktur von i2s erfolgen, aber auch von

c&m aus erfolgen, was bei der Betrachtung der Systemrollen von Bedeutung sein wird, wie im nächsten Kapitel gezeigt wird.

- **Der IT-Dienstleister ed.serv.** Das Unternehmen ed.serv wird in diesem Szenario vertreten durch die Ansprechpartner, die die Pflege der Serversysteme von c&m übernehmen. Hierfür existiert die Geschäftsrolle des „Service Managers“ auf Seiten von ed.serv. Diese Ansprechpartner besitzen die technische Kompetenz zur Pflege dieser Serversysteme.

- **Das IT-Consulting-Unternehmen ed.consult.** Als Beratungsunternehmen im IT-Bereich stellt ed.consult seinem Kunden c&m eine Menge an Mitarbeitern zur Verfügung, die c&m bei der geplanten Änderung der Zugriffskontrolle in der technischen Infrastruktur unterstützen. Aus diesem Grund kennt ed.consult in der Rolle des „Sicherheits-Consultant“ (engl. *security consultant*) die Software-Lösung ed.tec der Firma ed.soft und verfügt über die technische Kompetenz beim Aufbau einer rollenbasierten Zugriffskontrollarchitektur. Die Consultants stellen somit neben der Unterstützung von c&m auch den Vermittler zwischen c&m und ed.soft dar, um Erfahrungswerte im Umgang mit ed.tec an den Software-Hersteller weiterzuleiten.

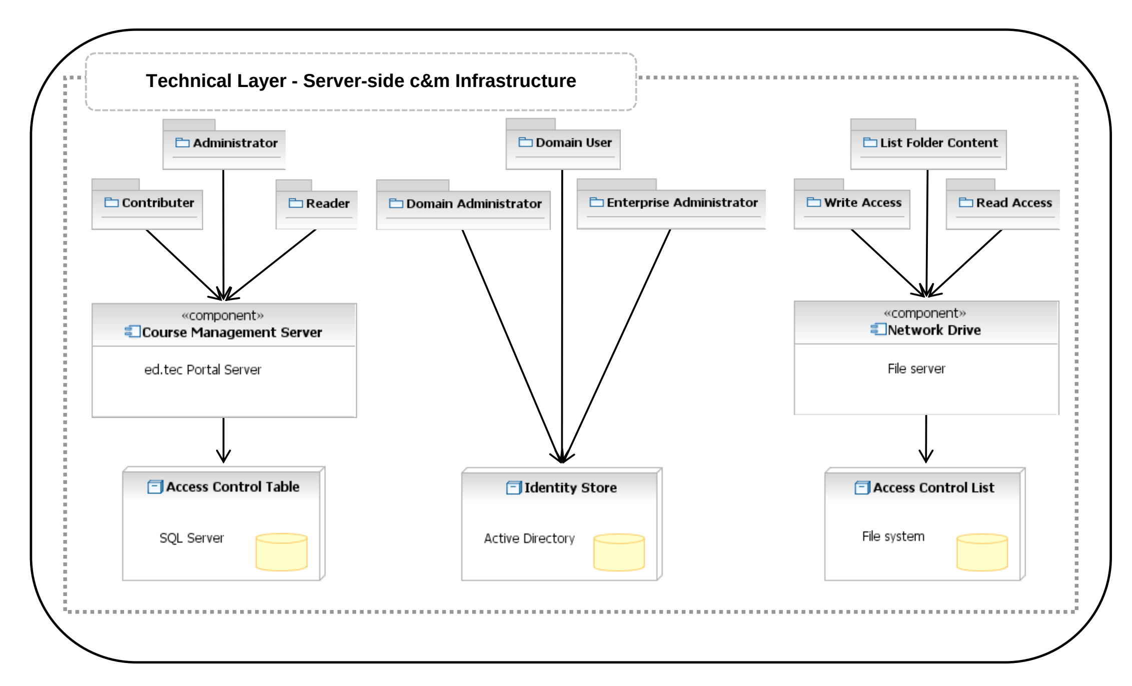
- **Der Software-Hersteller ed.soft.** Der Software-Hersteller ed.soft wird in diesem Szenario durch die Geschäftsrolle „Software-Entwickler“ (engl. *software developer*) vertreten, die maßgeblich an der Entwicklung und Verbesserung von ed.tec beteiligt ist. Dazu stehen Entwickler in dieser Rolle in direktem Kontakt mit den Sicherheits-Consultants von ed.consult.

7.1.2 Vorstellung der Systemrollen

An dieser Stelle werden die Systemrollen eingeführt, die in diesem Szenario auftreten. Es muss erwähnt werden, dass es sich dabei zunächst um atomare technische Berechtigungen handelt, was in BRBAC als System-Policy definiert wurden. Wie Information 39 verdeutlicht, besteht eine Systemrolle aus mindestens einer System-Policy als inhärentem Bestandteil, so dass die Begriffe System-Policy und Systemrolle an dieser Stelle äquivalent verwendet werden können. Die Vorteile, die in BRBAC aus der expliziten Trennung der Rollenbegriffe entsteht, wird in den Kapiteln 7.2.2 und 7.2.3 anhand des IST-Szenarios belegt. Da die Systemrollen mit den technischen Endsystemen in unmittelbarer Beziehung stehen, wird im Folgenden jeweils ein Endsystem und dessen Zweck im IST-Szenario eingeführt und anschließend die darin enthaltenen Systemrollen beschrieben.

Das Unternehmen c&m verfügt über einen Verzeichnisdienst, in dem alle Benutzerkonten angelegt sind, die innerhalb von c&m Zugriff auf die Endsysteme benötigen. Als Technologie wird in diesem Szenario Microsoft Active Directory eingesetzt, ein LDAP-konformer Verzeichnisdienst auf Basis des Windows Server 2003 Betriebssystems. In diesem Verzeichnis sind neben den Benutzerkonten der Mitarbeiter von c&m auch Benutzerkonten für das Unternehmen i2s, ed.consult und ed.serv angelegt, um den Beteiligten den Zugriff zu ermöglichen, der von ihnen benötigt wird. Die computergestützten Schulungen von c&m werden in Form der Intranet-Lösung ed.tec angeboten, einem Content-Management-System für Schulungen (engl. *course management system*), welches über eine eigene Berechtigungsstruktur verfügt und in Seiten angeordnet ist, um über einen Internet-Browser bedienbar zu sein. Dieses System verfügt über ein Modul zur Übersicht über die angebotenen Schulungen. Auch können sich Schulungsteilnehmer direkt über dieses Portal zu Schulungen an- bzw. abmelden und Bestätigungen über die Schulungen ausdrucken, an denen sie bislang teilgenommen haben. Als dritte Komponente bietet ed.tec ein Forum, in dem sich registrierte Benutzer austauschen und Administratoren Neuigkeiten bekanntgeben können. Ein drittes Unternehmenssystem stellt ein Dateiserver dar, über den Schulungsteilnehmer zusätzliche Schulungsmaterialien auch nach der Durchführung einer Schulung beziehen können. Der Zugriff auf dieses System wird über Zugriffskontrolllisten verwaltet,

die direkt mit den Identitäten der Benutzer verknüpft sind. Information 55 stellt die Endsysteme in Bezug zu exemplarischen Berechtigungen dar, auf die im Folgenden näher eingegangen wird.



Information 55: Case Study – Introduction to System Roles

Der zentrale Verzeichnisdienst Active Directory, den c&m einsetzt, bietet zur Zusammenfassung von Benutzern folgende drei Systemrollen an:

- Die Systemrolle „Domänenbenutzer" (engl. *domain user*) repräsentiert Benutzer mit eingeschränktem Rechteumfang innerhalb der technischen Infrastruktur von c&m. Domänenbenutzern ist es gestattet, sich innerhalb der Client-Infrastruktur von c&m anzumelden, wobei an dieser Stelle die Anmeldung nicht auf spezielle Client-Rechner eingeschränkt wird.

- Die Systemrolle „Domänenadministrator" (engl. *domain administrator*) gewährt neben dem uneingeschränkten Zugriff auf die Arbeitsplatzrechner auch den Zugriff auf die Serverbetriebssysteme von c&m. Diese Systemrolle hat daher insbesondere die Berechtigungen zur client- und serverseitigen Konfiguration der Betriebssysteme. In diesem Zusammenhang ist es möglich, Benutzerkonten zu verwalten, die im Verzeichnisdienst abgelegt sind.

- Die Systemrolle des „Unternehmensadministrators" (engl. *enterprise administrator*) baut auf der Systemrolle Domänenadministrator auf und erweitert dessen Kompetenzen um die Möglichkeit, Schemadefinitionen innerhalb von c&m zu ändern, wie es etwa bei der Integration zusätzlicher technischer Endsysteme der Fall sein kann. Diese Systemrolle verkörpert somit den größten Berechtigungsumfang innerhalb der Zugriffskontrolle von c&m.

Das zentrale Portal ed.tec, das das Unternehmen c&m einsetzt, verfügt über eine eigene Berechtigungsstruktur, die in einer Datenbank abgelegt ist. Aufgrund des Zugriffsmusters auf ein CMS-System werden folgende drei Systemrollen definiert:

- Die Systemrolle „Leser" (engl. *reader*) gewährt die Möglichkeit, den Inhalt innerhalb von ed.tec aufzulisten. Dabei wird an dieser Stelle von der Komplexität abstrahiert, dass sich der gesamte Inhalt von ed.tec in sehr viele verschiedene Seiten gliedert und der

Zugriff auf jede dieser Seiten individuell vorgegeben werden muss. Im Gegensatz zum klassischen lesenden Zugriff, wie er bei Dateisystemen vorkommt, verfügt die Systemrolle Leser zusätzlich über die Möglichkeit, bestehende Inhalte zu ergänzen. Jedoch sind Änderungen am Aufbau einer Seite, oder das Löschen von Inhalten, die nicht vom betreffenden Benutzer selbst erzeugt wurden, explizit ausgeschlossen. Dadurch wird realisiert, dass sich Benutzer, die diese Systemrolle besitzen, eigenständig zu Schulungen an- und abmelden können, Einträge im Forum vornehmen oder eigene Einträge selbständig widerrufen können.

- Mit der Systemrolle „Gestalter“ (engl. *contributer*) wird eine aktive Gestaltung einer Seite in ed.tec bezeichnet. Damit ist insbesondere gemeint, dass alte Inhalte gelöscht, neue hinzugefügt oder bestehende geändert werden können. Dies bezieht sich vor Allem auf die Verwaltung des Schulungsangebots von c&m. Anders als die Systemrolle Leser kann die Systemrolle Gestalter einen Überblick über alle Teilnehmer einer Schulung halten und diese selbständig nachträglich ein- oder austragen.

- Die dritte Systemrolle in ed.tec wird mit „Administrator“ bezeichnet und befähigt einen in dieser Rolle tätigen Mitarbeiter, systematische Einstellungen von ed.tec vorzunehmen, wie etwa die Benutzerverwaltung einhergehend mit der Änderung von Benutzerrechten. Nur dieser Rolle ist es gestattet, komplette Seiten zusammen mit dem Inhalt, der darauf publiziert ist, zu löschen.

Wie Information 55 aufzeigt, stellt c&m ein drittes Unternehmenssystem zur Verfügung, welches als unternehmensweite Datenablage verwendet wird. Hier befinden sich einerseits zusätzliche Materialien zu denjenigen Schulungen, die über ed.tec angeboten werden, aber auch c&m-interne Dokumente, wie etwa Verträge, Rechnungen oder Marketing-Material. Auch an dieser Stelle wird von der internen Struktur des Dateiservers abstrahiert und lediglich die Systemrollen eingeführt, die den Zugriff kanalisieren:

- Die Systemrolle „Ordnerinhalte anzeigen“ (engl. *list folder content*) steht für das Recht, den Inhalt eines Ordners anzuzeigen. Dies wird etwa benötigt, um einen Überblick über den Inhalt zu geben, ohne explizit das Recht zu gewähren, die darin abgelegten Dokumente zu öffnen.

- Die zweite Systemrolle „Leser“ (engl. *reader*) befähigt einen Benutzer, die Elemente eines Ordners zu öffnen. Diese Rolle beinhaltet somit das Recht, den Inhalt eines Ordners aufzulisten und die angezeigten Dokumente auch zu öffnen.

- Die dritte Systemrolle „Schreiber“ (engl. *writer*) ermöglicht das aktive Ändern vorhandener Inhalte, ebenso, wie das Hinzufügen oder das Löschen von Daten aus den betreffenden Ordnern.

Abschließend soll eine kurze Zusammenfassung dieses Teilkapitels gegeben werden. Zunächst wurde das Beispielszenario IST eingeführt und die darin beteiligten Unternehmen vorgestellt. Anschließend wurden die vorhandenen Geschäfts- und Systemrollen betrachtet. Auf dieser Grundlage wird nun im Folgenden die Tragfähigkeit des BRBAC-Modells gezeigt, welches durch das Vorgehensmodell aufgebaut wird.

7.2 Instanziierung von BRBAC in einem kommerziellen Rollenmanagementwerkzeug

7.2.1 Auswahl des Rollenmanagementwerkzeugs

In Kapitel 6 wurde die Bewertung zweier aktueller Rollenmanagementwerkzeuge vorgenommen und die wesentlichen Unterschiede zwischen ihnen herausgestellt. In diesem Teilkapitel

wird nun dasjenige der beiden ausgewählt, welche das Vorgehensmodell und das Rollenmodell BRBAC im vorgegebenen Szenario besser unterstützt. Die Auswahl wird im Folgenden begründet.

Das Vorgehensmodell aus Kapitel 5 modelliert die Entwicklung von Rollen als hybriden Prozess, der den *top-down*-Ansatz mit dem *bottom-up*-Ansatz kombiniert und so zur Entwicklung von Geschäfts- und Systemrollen führt. Diese explizite Unterscheidung des Rollenbegriffs stellt einen wesentlichen Beitrag von BRBAC dar. Aus diesem Grund sollte das ausgewählte Werkzeug sowohl ein hybrides Vorgehen, als auch eine Unterscheidung beim Rollenbegriff zulassen. Wie die Analyse in den Kapiteln 6.2 und 6.3 zeigt, wird momentan weder im Omada Identity Manager (OIM), noch im Sun Role Manager (SRM) explizit zwischen Geschäfts- und Systemrollen unterschieden, obwohl eine implizite Unterscheidung in beiden Werkzeugen dennoch möglich ist. Obwohl die konkrete Auslegung des Rollenbegriffs in beiden Werkzeugen durchaus unterschiedlich ist, bietet keines der Werkzeuge einen klaren Vorteil bei der technischen Umsetzung von BRBAC im Bezug auf den Rollenbegriff. Im Hinblick auf die Rechtevererbung entlang der Rollenhierarchien, die BRBAC modelliert, weist der Omada Identity Manager zwar eine bessere Unterstützung auf, jedoch ist dies für das Beispielszenario eher von untergeordneter Bedeutung.

Bereits aus Information 1 geht hervor, dass die Anzahl von Systemrollen in Unternehmen um bis zu zwei Größenordnungen größer ist, als die Zahl der Geschäftsrollen. In Anbetracht dieser Tatsache stellt vor allem die Modellierung von Berechtigungen bzw. daraus abgeleiteter Systemrollen eine sehr komplexe Aufgabe im *bottom-up*-Vorgehen dar. Im Vorgehensmodell wurde daher die Notwendigkeit nach Algorithmen zur Unterstützung bei dieser Aufgabe zum Ausdruck gebracht. Von einer derartigen Unterstützung profitieren somit insbesondere die Analyse- und Entwurfsphase des Vorgehensmodells. Der Omada Identity Manager bietet zum aktuellen Zeitpunkt keine Komponente für das *role engineering* an. Der Sun Role Manager hingegen verfügt für den Bereich *role engineering* über Algorithmen zur Entwicklung von Rollen. Diese Algorithmen werden in SRM insbesondere für Rollen mit technischen Details angewandt, was als Systemrolle aufgefasst werden kann.

Das Vorgehensmodell weist eine Phase mit Rückkopplung auf sich selbst auf, wodurch ein zirkulärer Porzess modelliert werden soll. Durch diese Phase kommen administrative Aufgaben eines Rollenmodells im Wirkbetrieb zum Ausdruck. Durch die explizite Modellierung des Rollenbetriebs wird nunmehr beachtet, dass Rollen im Wirkbetrieb laufend Änderungen unterliegen, mit denen in angemessener Weise umgegangen werden muss. Ein Rollenmanagementwerkzeug sollte daher administrative Aufgaben unterstützen, die im Wirkbetrieb entstehen. Hierbei bieten beide Werkzeuge in gleicher Weise Prozesse an, obwohl automatische Arbeitsabläufe in OIM genauer spezifiziert werden können. Im Rahmen dieser Fallstudie bietet aber keines der Werkzeuge einen wesentlichen Vorteil.

Zusammengefasst liegen drei wesentliche Kriterien zur Werkzeugauswahl zugrunde:

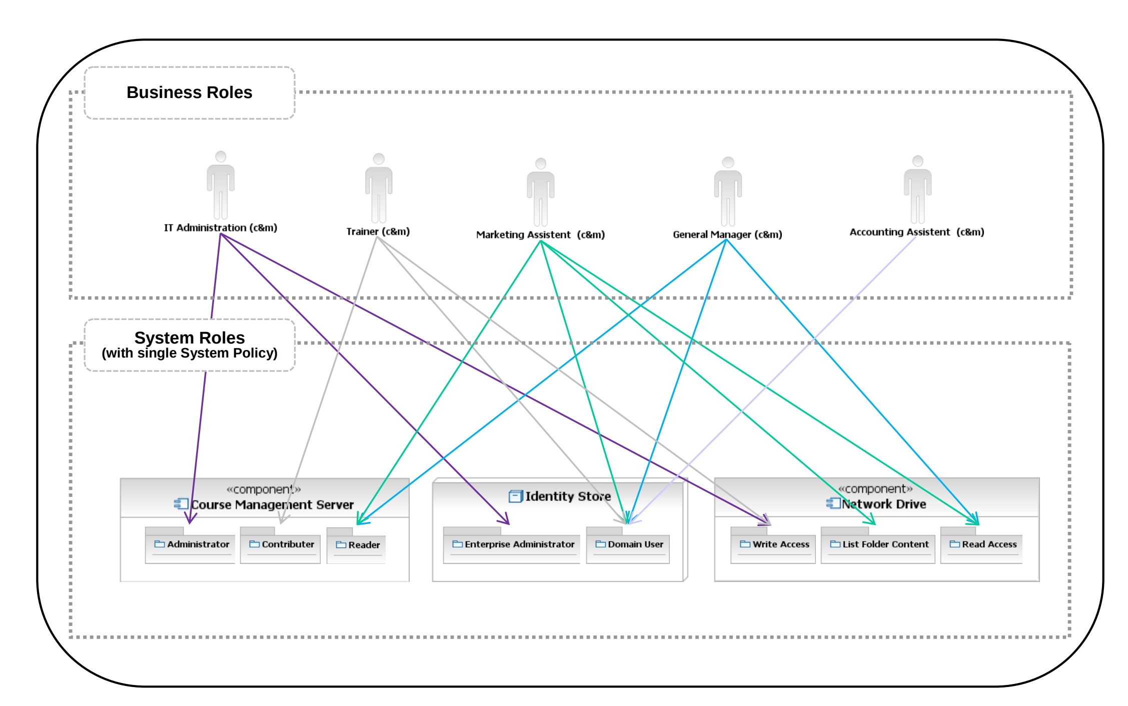
- Die Möglichkeit, Geschäftsrollen und Systemrollen zu definieren und zumindest implizit zu unterscheiden.
- Die Unterstützung eines hybriden Vorgehens bei der Entwicklung von Rollen.
- Die algorithmische Unterstützung bei der Systemrollenentwicklung.
- Die Unterstützung administrativer Aufgaben in Form von *workflows* für die Betriebsphase

Diese Kriterien werden in der vorliegenden Version vom Sun Role Manager besser unterstützt, so dass im Folgenden dieses Werkzeug zur Umsetzung von BRBAC im IST-Szenario verwendet wird.

7.2.2 Rollenanalyse und Rollenentwurf

Nachdem im vorangegangenen Kapitel ein Rollenmanagementwerkzeug für die Umsetzung von BRBAC ausgewählt wurde, werden die im IST-Szenario definierten Rollen nun im Sun Role Manager umgesetzt. Dabei werden die Geschäfts- und Systemrollen aus Kapitel 7.1 aufgegriffen und miteinander in Beziehung gesetzt. Das Ergebnis dieses Entwurfs ist in Information 56 dargestellt. Für eine vereinfachte Darstellung ist an dieser Stelle lediglich ein Ausschnitt gewählt worden, der die Geschäftsrollen von c&m enthält. Das Gesamtbild ist unter Anhang C in Information 65 aufgeführt.

Wie das Vorgehensmodell spezifiziert, wird in der Analysephase im Hinblick auf die entstehenden Rollen zunächst die Ist-Situation im Unternehmen erfasst und modelliert. Die Vorstellung des IST-Szenarios sowie die Formulierung der Geschäfts- und Systemrollen im Bezug zu den technischen Systemen, in denen sie definiert wurden, kann somit als Ergebnis der Analysephase angesehen werden und die Abbildungen aus Kapitel 7.1 als grobe Spezifikation, die Teil der Artefakte sind, die in die Entwurfsphase übergeben werden. Wie aus der verbalen Formulierung aus Kapitel 7.1 auch hervorgeht, verfügen die Geschäftsrollen über unterschiedliche Rechte innerhalb der technischen Infrastruktur von c&m, was im Folgenden konkretisiert werden soll. Ergänzt werden nun die Systemrollen, die die erwähnten Geschäftsrollen im IST-Szenario aufweisen.



Information 56: Case Study – Role Mapping with Simple System Roles

In Information 56 sind die erteilten Systemrollen in Bezug zu den Geschäftsrollen dargestellt. Durch die farbliche Markierung wird deutlich, welche Geschäftsrollen mit den Systemrollen verknüpft sind. Diese Verknüpfungen werden im Folgenden begründet.

Der Sicherheits-Consultant steht in direktem Kontakt mit der technischen Fachabteilung von c&m und unterstützt diese bei technischen Änderungen. Aus diesem Grund benötigt diese Rolle administrativen Zugriff auf die c&m-Infrastruktur. Dies betrifft einerseits den Verzeichnisdienst, aber auch das CMS-System ed.tec. Der dritte Dienst, der Netzwerkfreigaben ermöglicht, ist für den Consultant nicht von Belang, so dass hierfür keinerlei Rechte vergeben werden müssen. Der Service-Manager der Firma ed.serv kümmert sich um die Pflege der Serversysteme von c&m, benötigt also das Recht, sich an diesen Systemen anmelden zu können. Ihm soll es allerdings nicht gestattet sein, die Dienste von c&m zu nutzen oder Einblick erhalten zu können,

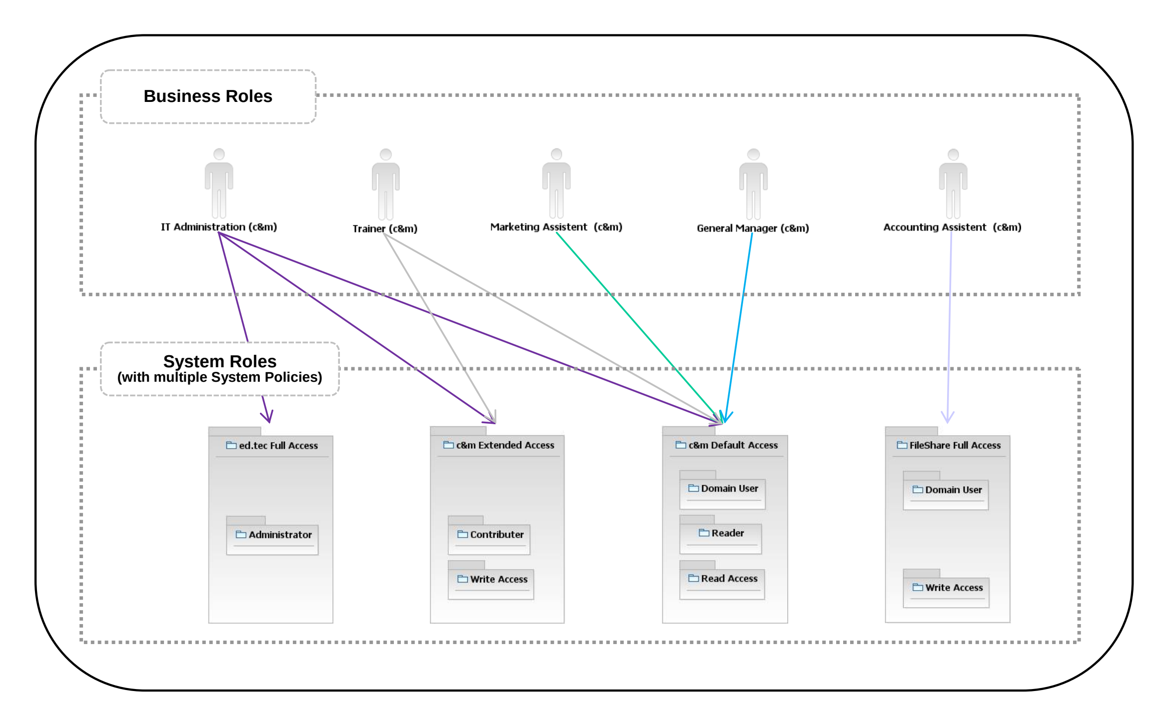
weswegen ihm weder Rechte auf ed.tec, noch auf den Dateiserver eingeräumt werden. Den Geschäftsrollen von i2s, `General Manager (i2s)` sowie `Course participant (i2s)` wird die Möglichkeit eingeräumt, am Schulungsangebot von c&m teilzunehmen. Dies bedeutet, dass sie Zugriff auf ed.tec sowie den Dateiserver benötigen, wo zusätzliche Daten zu den angebotenen Schulungen abgelegt sind. Aus diesem Grund benötigen beide Geschäftsrollen auch Benutzerkonten in der c&m-Infrastruktur. Gleiches gilt für die Geschäftsrolle `General Manager (c&m)`. Es sei an dieser Stelle angemerkt, dass diese Rolle in der c&m-Infrastruktur weit mehr Rechte besitzt, da dies allerdings außerhalb des betrachteten IST-Szenarios liegt, werden diese Rechte nicht explizit aufgeführt. Die Geschäftsrolle `Accounting Assistent (c&m)` verwendet den Dateiserver zur Ablage von buchhalterischen Dokumenten wie etwa Verträge mit den Kunden oder Rechnungen von ed.serv bzw. ed.consult. Die Marketing-Abteilung von c&m, vertreten durch die Geschäftsrolle `Marketing Assistent (c&m)` befasst sich damit, das Schulungsangebot an die Kundenwünsche anzupassen. In diesem Sinne benötigt sie Zugriff auf ed.tec und den Dateiserver, um die Einteilungen zu Schulungen zu überwachen und das Interesse an zusätzlichen Schulungsmaterialien zu erkennen. Die Geschäftsrolle `Trainer (c&m)`, die Schulungen durchführt, muss dazu berechtigt sein, Inhalte auf ed.tec zur Verfügung zu stellen und neue Schulungen einzutragen, um Teilnehmer darüber in Kenntnis zu setzen. Aus diesem Grund wird diese Rolle mit der Systemrolle `Contributer` verknüpft. Zur Ablage von ergänzendem Material bereits abgehaltener Schulungen ist es daher nötig, schreibend auf den Dateiserver zugreifen zu können. Die Geschäftsrolle `IT Administration (c&m)` verfügt über den Vollzugriff auf die gesamte technische Infrastruktur von c&m, da sie neben der Administration der Benutzerkonten auch die Administration der Serversysteme verantwortet.

7.2.3 Das Rollenmodell BRBAC im Beispielszenario

Bei dem hier gewählten Szenario IST handelt es sich lediglich um einen kleinen und beschränkten Ausschnitt aus dem Gesamtkontext eines Unternehmens. Dabei wurden insgesamt neun Geschäftsrollen sowie drei Unternehmenssysteme eingeführt, die jeweils über drei technische Berechtigungen, oder System-Policys verfügen, wobei im hier gewählten Ausschnitt lediglich fünf dargestellt wurden. Eine gültige Umsetzung von BRBAC wurde im vorangegangen Kapitel aufgezeigt, in dem für jede System-Policy eine eigene Systemrolle konzipiert wurde, die die darin enthaltene Berechtigung in sich kapselt. Bereits dadurch wird die direkte Bindung an das technische Endsystem aufgehoben, was der Forderung von BRBAC nach einer Abstraktion in den Systemrollen von den technischen Endsysteme nachkommt. Diese Abstraktion kann sogar noch klarer gezeigt werden: Hierfür werden Algorithmen zum *role-mining* auf den Datenbestand der Systemrollen angewandt und der *bottom-up*-Ansatz weiterhin verfolgt. Es wird nunmehr versucht, Zugriffsmuster in den bisherigen Systemrollen zu erkennen und die System-Policys in den bisherigen Systemrollen zu allgemeinen Systemrollen zusammengefasst. An dieser Stelle sei verweisen auf die Modellierungsaspekte generische Rollen und Policy-Erweiterungen, wie etwa die Verwendung von Platzhaltern oder die Rechtevererbung von Policys. Diese Aspekte kommen insbesondere in großen Szenarien zum Tragen und werden im Rahmen des IST-Szenarios nicht näher betrachtet, weil es hierfür nicht komplex genug ist. Die Verwendung von statischem SoD ist durch die Verwendung von expliziten Geschäfts- und Systemrollen, die sich wechselseitig ausschließen können trivial: Es werden hierfür Rollen definiert, die sich gegenseitig ausschließen. Dabei kann nun explizit zwischen sich wechselseitig ausschließenden Geschäfts- und Systemrollen unterschieden werden. Auch dieser Aspekt wird nicht näher betrachtet. Wie dieses Kapitel zeigen wird, führt BRBAC alleine durch die Trennung von Rollen bereits zu einer deutlichen Komplexitätsreduktion.

Durch Information 56 wird die Komplexität bei der Verknüpfung von Rollen bereits deutlich, obwohl in diesem Szenario lediglich neun Geschäfts- und neun Systemrollen vorkommen. Ohne die farbliche Markierung der Beziehungen würde bereits dieser kleine Teilausschnitt eines Unternehmens undeutlich werden. Es soll nun das Potential des Systemrollenkonzepts von BRBAC verdeutlicht werden. Dazu werden Muster in den Zugriffen analysiert und Systemrol-

len konzipiert, die System-Policys aggregieren. Im hier gewählten Beispiel lässt sich etwa erkennen, dass sich ein eingeschränkter Zugriff auf die c&m-Infrastruktur dadurch auszeichnet, dass er die System-Policys `Domain User`, `Reader (ed.tec)` und `Read Access` auf den Dateiserver besitzt. Führt man eine exemplarische Zusammenfassung der System-Policys konsequent fort, führt dies zur Bildung von fünf Systemrollen. Das resultierende Rollenmodell BRBAC ist in Information 57 dargestellt, wobei hier derselbe Ausschnitt gewählt wurde, wie in der vorherigen Abbildung. Der Gesamtausschnitt ist in Information 66 in Anhang C dargestellt.



Information 57: Case Study – Role Mapping with Complex System Roles

Vergleicht man nunmehr Information 56 und Information 57, so stellt man fest, dass BRBAC durch die Aggregation von System-Policys deutlich klarer strukturiert ist. Im Gegensatz zu neun Systemrollen im Gesamtbild der direkten Konvertierung von System-Policys existieren nunmehr lediglich fünf Systemrollen. In dieser Abbidung sind davon lediglich vier dargestellt. Dies bedeutet eine Verringerung bei den Systemrollen um etwa 44%. Zwar ist dies in Anbetracht von lediglich neun System-Policys kein statistisch aussagekräftiger Wert, jedoch stellt das IST-Szenario eine durchaus realistische Fallstudie aus einem unternehmerischen Kontext dar. Da die einzelnen Policys stets diskrete Zugriffsberechtigungen in sich kapseln, kann davon ausgegangen werden, dass sich die Vorteile von BRBAC insbesondere in großen Einrichtungen potenzieren.

Eine weitere Sichtweise stellt die Betrachtung der Zuweisungen statt der Rollenanzahl dar: Die Zahl der Systemrollenzuweisungen sinkt in diesem Fall von 24 auf 13 gemessen am Gesamtbild. Hierdurch wird die Verringerung der Komplexität ebenfalls deutlich: Die Verringerung der Systemrollenzuweisungen sinkt um etwa 54%. Dies ist in etwa in der Größenordnung der Verringerung an Rollen, weswegen davon ausgegangen werden kann, dass BRBAC durch die explizite Rollentrennung zu einer deutlichen Komplexitätsreduktion im Wirkbetrieb führt.

7.2.4 Betriebsphase von BRBAC im Sun Role Manager

Nach der Betrachtung der Geschäfts- und Systemrollen aus BRBAC sowie deren Umsetzung im Sun Role Manager soll nun zum Abschluss der Fallstudie ein Einblick in die Betriebsphase des Vorgehensmodells gegeben werden. Wie in Kapitel 5.5 definiert wurde, betrachtet diese Phase administrative Aufgaben, die beim Betrieb von BRBAC vorhanden sind. Dies soll nun im IST-Szenario beleuchtet werden. Zur Veranschaulichung zeigt Information 58 die grafische Benut-

zeroberfläche im Sun Role Manager für die Rolle `IT Administration (c&m)`. Die administrativen Aufgaben im Wirkbetrieb von BRBAC können unterteilt werden in die Aufgabenbereiche „Rollenverwaltung“, „Prozess-Überwachung“ und „Policy-Verletzungen“. Diese drei Bereiche werden anhand eines Beispiels erläutert und in Bezug gesetzt zu Information 58.



Information 58: Case Study – Role Operation in Sun Role Manager

- **Rollenverwaltung.** Dieser Arbeitsbereich umfasst sämtliche Aufgaben, die im Zusammenhang mit Rollen auftreten. Dies beginnt mit der Einteilung von Benutzern in Geschäftsrollen, der Einteilung von System-Policys in Systemrollen sowie der Zuweisung von Geschäfts- zu Systemrollen. In der bisherigen Schilderung der Fallstudie verfügt die Rolle `Security Consultant (ed.consult)` über administrative Rechte auf die Server- und Client-Systeme von c&m, nicht jedoch über Rechte auf das Schulungsportal ed.tec. Wenn ed.consult im Rahmen einer geplanten Systemänderung an ed.tec aktiv eingebunden werden soll, ist es nötig, dass dieser Rolle die entsprechenden Zugriffsrechte erteilt werden. Um dies zu bewerkstelligen, muss die Geschäftsrolle `Security Consultant (ed.consult)` zusätzlich in die Systemrolle `Full Access (ed.tec)` eingeteilt werden.

- **Prozess-Überwachung.** Dieser Arbeitsbereich befasst sich mit den Prozessen, die innerhalb des Sun Role Manager definierbar sind. Im Rahmen der automatischen *workflows* ist es beispielsweise möglich, einen Prozess zu definieren, der die IT Administration von c&m darüber in Kenntnis setzt, dass eine Geschäftsrolle zusätzliche Rechte benötigt. Dazu verfügt der Sun Role Manager über eine grafische Übersicht mit ausstehenden und bereits erledigten Anfragen (engl. *pending requests*, *completed requests*). Auch werden für die erteilten Policys zwei eigene Ansichten angeboten (engl. *my certifications*, *certify/revoke statistics*).

- **Policy-Verletzungen.** Dieser letzte hier beleuchtete Arbeitsbereich befasst sich mit Rechteüberschreitungen eines Benutzers, die daraus resultieren, dass dieser Benutzer in der Ausübung seiner Geschäftsrolle den darin spezifizierten Berechtigungsumfang überschreiten. Im Beispiel könnte der Zugriffsversuch des Sicherheits-Consultant auf das ed.tec-Portal erkannt werden und in der Ansicht für die Policy-Verletzungen angezeigt werden (engl. *identity audit policy violation*). Auf diese Weise würde die IT-Administration von c&m direkt darüber informiert, dass der Policy-Umfang der Rolle nicht ausreicht und für die Dauer der geplanten Systemänderung von ed.tec erweitert werden muss.

7.3 Resümee

In diesem Kapitel wurden die Rollenmodelle, die in dieser Arbeit entwickelt wurden, in einem Beispielszenario angewandt. Dadurch wurde die Tragfähigkeit sowie die Relevanz dieser Arbeit demonstriert. Die wesentlichen Inhalte dieses Kapitels sollen nun kurz zusammengefasst werden.

Zunächst wurde ein Beispielszenario eingeführt, welches mehrere Unternehmen aufweist, die miteinander in Beziehung stehen. Neben der Erwähnung der Geschäftsmodelle dieser Unternehmen wurde ein Ausschnitt der existierenden Geschäftsrollen gegeben sowie eine Menge an technischen Systemen zusammen mit darin definierten Berechtigungen vorgestellt. Auf dieser Grundlage wurde in Kapitel 7.2.1 mit dem Sun Role Manager ein Rollenmanagementwerkzeug bestimmt, in welchem BRBAC gemäß dem dafür definierten Vergehen umgesetzt werden sollte. Insbesondere die Möglichkeit, Geschäfts- und Systemrollen – wenn auch nur eingeschränkt – formalisieren zu können, die Unterstützung eines hybriden Vorgehens zusammen mit *role-mining*-Algorithmen im Rollenentwurf und die Unterstützung der Betriebsphase von BRBAC haben zur Auswahl des Sun Role Manager geführt. Im Anschluss daran folgte in Kapitel 7.2.2 eine genauere Betrachtung der Geschäfts- und Systemrollen. Durch die explizite Modellierung von Systemrollen und eine exemplarische Anwendung von *role-mining*-Prinzipien konnten sich die Systemrollen sogar noch zusammenfassen lassen und somit die Verringerung der Komplexität von BRBAC belegt werden. Die explizite Trennung von Geschäfts- und Systemrollen stellt die primäre Stufe der Komplexitätsreduktion und die Konzeption von Systemrollen, die mehr als nur eine System-Policy beinhalten die sekundäre Stufe der Komplexitätsreduktion dar. Beides konnte in Kapitel 7.2.3 belegt werden. Abschließend wurde die hohe Relevanz einer eigens modellierten Betriebsphase verdeutlicht, was durch ein Beispiel belegt werden konnte.

Um die in dieser Arbeit entwickelten Modelle und Konzepte anschaulich beschreiben zu können, ist das gewählte Szenario auf einen spezifischen Aspekt beschränkt. Es dürfte daher höchst interessant sein, wie sich BRBAC in der wirtschaftlichen Umsetzung in einem großen Szenario verhält, insbesondere, wie sich die hier entwickelten Prinzipien zu SoD, generischen Rollen und Policy-Erweiterungen auswirken. Es kann zu diesem Zeitpunkt lediglich ein Indizienbeweis geliefert werden, dass all diese Prinzipien den zentralen Zielen von BRBAC entsprechen, wie aus Kapitel 7.2.3 hervorgeht. Als zusätzlicher Beleg wird auf [Ke02] verwiesen, wo ähnliche Vereinfachungen eingeführt und die Tragfähigkeit durch Berichte aus praktischen Umsetzungen belegt wurden.

8 ZUSAMMENFASSUNG UND AUSBLICK

8.1 Ergebnisse dieser Arbeit

In diesem Kapitel sollen die zentralen Ergebnisse der vorliegenden Arbeit zusammengefasst und miteinander in Bezug gestellt werden.

Zunächst wurden in Kapitel 2 die Grundlagen für eine detaillierte Diskussion von Modellen für die rollenbasierte Zugriffskontrolle gelegt, ausgehend von einer Betrachtung der Subjekt/Objekt-Relation, die als Basis von Zugriffskontroll-Architekturen angesehen werden kann. Diese Relation wurde in Bezug gesetzt zum Paradigma der rollenbasierten Zugriffskontrolle, die das Konzept „Rolle" als Indirektionsstufe zwischen dem zugreifendem Subjekt und Objekt als Ziel des Zugriffs einführt. In der Folge wurden die vier Modelle des NIST-RBAC-Standardrahmenwerks vorgestellt und jeweils die Modellspezifikation sowie die funktionale Spezifikation erläutert. Ein wichtiges Merkmal von Zugriffskontrollarchitekturen ist die Möglichkeit, wechselseitigen Ausschluss von Rechten spezifizieren zu können. Dieses als *separation of duty* bekannte Prinzip wird in der rollenbasierten Zugriffskontrolle durch Rollen realisiert, die sich wechselseitig ausschließen. Hierbei wird zwischen statischem und dynamischem Ausschluss unterschieden, wobei durch den statischen wechselseitigen Ausschluss zwei Rollen ohne Beschränkungen als unvereinbar angesehen werden, die im dynamischen Fall nur dann unvereinbar sind, wenn im gleichen Kontext agiert wird. Dies beendete die Betrachtung der theoretischen Grundlagen. Den Abschluss bildete die Vorstellung typischer Aufgaben von Rollenmanagementwerkzeugen, was die Grundlage für eine eingehende Auseinandersetzung mit zwei Vertretern dieser Werkzeugkategorie darstellte.

Auf dieser Grundlage wurde dann in Kapitel 3 der aktuelle Stand in Wissenschaft und Technik in den Bereichen der rollenbasierten Zugriffskontrolle und des Rollenmanagements dargelegt. Im Bereich der Wissenschaft wurde mit dem ERBAC-Modell sowie den Erweiterungen dazu ein aktuelles Rollenmodell vorgestellt, welches sich mit den Problemen befasst, die aus praktischen Umsetzungen von NIST-RBAC im Unternehmenskontext resultieren. Das ERBAC-Modell stellt unmittelbar die Grundlage des Rollenmodells BRBAC dar, welches in dieser Arbeit entwickelt wurde. Des Weiteren wurde ein Modell zur algorithmengestützten Entwicklung von Rollen vorgestellt, das Techniken aus dem Bereich *data-mining* auf den Datenbestand eines Unternehmens anwendet. Diese Herangehensweise wird mittlerweile auch in kommerziellen Rollenmanagementwerkzeugen verwendet. Den Abschluss aktueller wissenschaftlicher Forschungsergebnisse stellte ein Prozessmodell dar, welches sich mit dem Lebenszyklus von Rollen befasst. In diesem Modell werden zwei separate Phasen vorgeschlagen, die sich mit administrativen Aufgaben eines Rollenmodells im produktiven Einsatz beschäftigen. Die beiden letztgenannten Forschungsbeiträge bildeten die Basis des in dieser Arbeit entwickelten Vorgehensmodells. Nach der Betrachtung aktueller wissenschaftlicher Forschungsergebnisse wurden zwei Rollenmanagementwerkzeuge vorgestellt als Beleg aktueller Betätigungen in der Industrie. In analoger Weise wurde für den Identity Manager der Firma Omada und den Role Manager der Firma Sun zunächst deren Architekturen und Datentypen vorgestellt und die Komponenten genannt, über die diese Werkzeuge verfügen. Auf diese Weise wurde der Bezug hergestellt zu den typischen Aufgaben von Rollenmanagementwerkzeugen aus dem Grundlagenkapitel. Nach diesem Überblick folgte für beide Werkzeuge dann eine detaillierte Vorstellung der einzelnen Arbeitsbereiche.

Den ersten zentralen Aspekt der vorliegenden Arbeit stellte die Entwicklung des Rollenmodells BRBAC (engl. *business role-based access control*) dar. Durch diese Namensgebung kommt der Bezug zu praktischen Umsetzungen rollenbasierter Zugriffskontrollarchitekturen im Unternehmenskontext unmittelbar zum Ausdruck. Wie [Ke02] bereits belegt, existieren gerade durch den starken Einsatz unterschiedlicher Technologien auf technischer Ebene in Unternehmen eine Vielzahl an Rollen und zumindest implizit unterschiedliche Rollentypen. Diese Tatsache stellte

die Forderung an ein Rollenmodell nach einer möglichst effizienten Verwendung von Rollen sowie gleichermaßen nach einer Reduktion von Komplexität. Anders als ERBAC sieht diese Arbeit die explizite Trennung von geschäftsnahen und technischen Rollen als zentrale Anforderung. Eine Rolle weist dabei eine Menge an Policys auf, die ihren Zugriffsumfang spezifiziert und diesen auf ein Zugriffsziel festlegt. Zusätzlich zur Spezifikation von Geschäfts- und Systemrollen wurden generische Rollen als zweite Modellspezifikation eingeführt. Diese Spezialform von Geschäfts- und Systemrollen abstrahiert vom konkreten Zugriffsziel und somit lässt sich ein Berechtigungsumfang spezifizieren, der in unterschiedlichen Kontexten gleichermaßen angewandt werden kann. Im Folgenden wurde auf die Modellierung von wechselseitigem Ausschluss (engl. *separation of duty*, SoD) eingegangen und eine Möglichkeit aufgezeigt, wie neben statischem SoD insbesondere dynamisches SoD modelliert werden kann, was in bisherigen Rollenmodellen für den Einsatz in Unternehmen nicht beachtet wurde. Das BRBAC-Modell verfolgt dabei den Ansatz, dass der für dynamisches SoD notwendige Kontext im Benutzerobjekt als eindeutigem Identifikator enthalten sein kann. Nach der Spezifikation von Rollen wurde anschließend auf die Modellierung von Policy-Erweiterungen eingegangen. Auch hierbei war das Ziel, ein möglichst kompaktes und effizientes Rollenmodell zu entwerfen. Die erste Erweiterung betraf die Verwendung von Platzhalterwerten für Attribute in Policys. Dadurch wird es möglich, Policys für konkrete Zielsysteme zu definieren, die den Berechtigungsumfang offen lassen. Dieser wird erst durch ein Attribut des Benutzerobjekts zum Zeitpunkt des Zugriffs festgelegt. Eine zweite Erweiterung greift an der Hierarchie von Rollen an und spezifiziert Möglichkeiten, die Vererbung von Policys, die mit der Hierarchiebildung einhergeht, zu steuern. Diese Prinzipien sind in identitätsbasierten Verzeichnisdiensten bereits vorhanden und wurden auf BRBAC angewandt. Durch das Attribut *block policy inheritance* in einer Policy wird erreicht, dass sie explizit von der Vererbung ausgeschlossen wird um eine spezifische Berechtigung nicht an tieferliegende Hierarchiestufen weiterzugeben, während das Attribut *no override* angibt, dass der in der Policy definierte Berechtigungsumfang durch zusätzliche Policys in der Hierarchie nicht überschrieben werden kann. Eine dritte Erweiterung auf Policy-Ebene stellen Joker-Policys dar. Diese stehen ausschließlich für Systemrollen zur Verfügung, weil diese Erweiterungen im direkten Bezug zu technischen Endsystemen stehen. Durch die Joker-Policy wird ausgedrückt, dass das Vorhandensein von Systemrollen in einer Geschäftsrolle davon abhängen kann, in welchem Kontext sich die Geschäftsrolle gerade befindet. Befindet sie sich aktuell in einem Kontext, der die Bedingung nicht erfüllt, die durch die Joker-Policy festgelegt ist, wird dem Benutzer diese Systemrolle im vorliegenden Kontext entzogen. Diese Betrachtung beendete die Entwicklung des Rollenmodells BRBAC, welches zum Schluss in seiner Gesamtheit dargestellt wurde und die bisher einzeln betrachteten Aspekte in Kombination beleuchtet wurden.

Nach der Entwicklung des Rollenmodells wurde in Kapitel 5 ein Vorgehensmodell entwickelt, welches das Rollenmodell BRBAC in den Wirkbetrieb überführt. Ein zentrales Ziel war es daher, dass sich die in BRBAC geforderte Trennung von Geschäfts- und Systemrollen auch im Vorgehen widerspiegelt. Eine zweite Forderung stellte die explizite Beachtung des Lebenszyklus von Rollen dar. Um dieses Ziel zu erreichen, wurde ein hybrides Vorgehen entwickelt, dass bei der Entwicklung von Geschäftsrollen anders vorgeht, als bei Systemrollen. Bei der Entwicklung von Geschäftsrollen wird gemäß dem *top-down*-Vorgehen von einer abstrakten Sicht auf ein Unternehmen ausgegangen und diese Sicht konkretisiert, um daraus geschäftliche Aufgaben zu identifizieren, die in Geschäftsrollen münden. Beim Entwurf von Systemrollen wird gemäß dem *bottom-up*-Vorgehen bei der Analyse der existierenden technischen Einzelberechtigungen in den Endsystemen des Unternehmens begonnen und diese durch *role-mining*-Algorithmen zur Mustererkenung in Systemrollen zusammengefasst. Das Vorgehen orientiert sich am klassischen Entwicklungszyklus von Software und ist in vier Phasen unterteilt. Für jede Phase wurden definiert, was in dieser Phase zu erledigen ist, was das Ziel der Phase darstellt bzw. welche Zielartefakte erzeugt werden, wer an der Phase beteiligt ist und welche Gründe für Rückkopplungen zu früheren Phasen sprechen. Das Ziel der Analyse- und Entwurfsphase ist der Entwurf von Geschäfts- und Systemrollen, was durch das hybride Vorgehen parallel geschehen kann. Den Abschluss der Entwurfsphase bildet die Spezifikation dieser beiden Rollentypen zusammen

mit der Geschäftsrollen/Systemrollen-Relation, die diese beiden Rollentypen verbindet. Zusammen mit diesen Spezifikationen dient ein Pflichtenheft als Zielartefakt dieser Phase. Zur Implementierung in der Implementierungsphase muss zunächst das Rollenmanagementwerkzeug selbst im Unternehmen installiert und konfiguriert werden, ehe das spezifizierte Rollenmodell in diesem Werkzeug umgesetzt werden kann. Das Vorgehen legt gerade für größere Einrichtungen eine schrittweise Implementierung nahe, begleitet von einem Testbetrieb, um Erfahrungswerte im Umgang mit der neuen Zugriffskontrollarchitektur ins Rollenmodell zurückführen zu können. Durch die Sammlung von Erfahrungswerten wird auch sichergestellt, dass die in der Analyse- und Entwurfsphase konzipierte Rollenmodellinstanz den Gegebenheiten des Unternehmens entspricht, was die Akzeptanz der Modells im Wirkbetrieb erhöht. Die letzte Phase des Vorgehensmodells befasst sich zentral mit der Pflege des Rollenmodells im Wirkbetrieb und kommt somit der Forderung nach, dass das Vorgehensmodell als Lebenszyklus aufgefasst werden kann. Auch in diesem Kapitel wurden die zentralen Aspekte kurz zusammengefasst.

Das Ziel von Kapitel 6 war die Bewertung der Rollenmanagementwerkzeuge, die in Kapitel 3 bereits eingeführt wurden. Dazu wurde zunächst ein Kriterienkatalog entwickelt und eine Metrik für die Kriterien bestimmt. Dabei wurden solche Kriterien, die für den Einsatz von BRBAC wichtig waren ebenso ausgewählt, wie solche, die das Werkzeug in seiner Gesamtheit bewerten sollten. Das erste Kriterium „Rollen" befasste sich mit der Unterstützung von Geschäfts- und Systemrollen im vorliegenden Werkzeug. Das zweite Kriterium „Hierarchien" bewertete den Unterstützungsgrad für Hierarchien, wobei auch hier die beiden Rollentypen differenziert betrachtet wurden. Das dritte Kriterium bewertete die Unterstützung von *role-mining*-Algorithmen zur Entwicklung von Rollen. Mit der Bewertung von automatischen Arbeitsabläufen (engl. *workflows*) wurde die Unterstützung für administrative Aufgaben im produktiven Einsatz bewertet. Diese Kriterien standen in unmittelbarer Relation zu den Rollenmodellen, die in den Kapiteln 4 und 5 entwickelt wurden. Zur querschnittlichen Bewertung der Werkzeuge wurden noch die Unterstützung von dienstorientierten Architekturen und der automatischen Rückführung von Rollen und Rolleninformationen in die technischen Endsysteme des Unternehmens bewertet und abschließend eine subjektive Einschätzung der Benutzerfreundlichkeit gegeben. Es ließ sich feststellen, dass der Omada Identity Manager Vorteile bei der Unterstützung von *workflows* besitzt sowie bei denjenigen Kriterien, die nicht in unmittelbarem Bezug zu den Rollenmodellen dieser Arbeit stehen und der Sun Role Manager durch die *role-mining*-Komponente sowie dessen Rollenmodell Vorteile aufweist. Dies wurde in einem abschließenden Fazit herausgestellt.

Abschließend wurde in Kapitel 7 die technische Umsetzung von BRBAC gemäß dem Vorgehensmodell in einem der beiden Werkzeuge durchgeführt. Dazu wurde zunächst ein Beispielszenario und die darin definierten Geschäfts- und Systemrollen vorgestellt. Anschließend wurde mit dem Sun Role Manager dasjenige Werkzeug ausgewählt, das in diesem Szenario für die Umsetzung besser geeignet ist und anschließend auf die Implementierung der Rollen aus BRBAC eingegangen. Auch wurden die Vorteile der expliziten Trennung anhand des Szenarions belegt und illustriert. Den Abschluss dieses Kapitels bildete eine Betrachtung der Rollenbetriebsphase sowie die vom Sun Role Manager hierfür zur Verfügung gestellten Komponenten.

8.2 Ausblick auf zukünftige Forschungsarbeiten

In diesem abschließenden Kapitel sollen Anknüpfungspunkte für weitere Forschungen aufgezeigt werden. Mit BRBAC wurde ein Rollenmodell entwickelt, das sich durch die explizite Trennung der Rollentypen bislang nicht eindeutig in einem bestehenden Werkzeug abbilden lässt. Aus diesem Grund liefert dieses Modell viel Spielraum für weitere Arbeiten in diesem höchst aktuellen Forschungsgebiet.

Mit BRBAC wurde ein Rollenmodell entwickelt, das explizit zwischen zwei Rollentypen unterscheidet. Wie die Werkzeugbewertung aus Kapitel 6 aufgezeigt hat, ist diese Unterscheidung

zum aktuellen Zeitpunkt lediglich implizit möglich. Es dürfte daher höchst interessant sein, welche Möglichkeiten eine technische Implementierung eines Rollenmanagementwerkzeugs bietet, das auf BRBAC basiert.

Durch das Fehlen einer expliziten Rollentrennung ist es nicht möglich gewesen, viele der in BRBAC definierten Mechanismen einer Bewertung zu unterziehen. Insbesondere für die Policy-Mechanismen kann erst in einem BRBAC-konformen Rollenmanagementwerkzeug das Ausmaß der Komplexitätsreduktion gezeigt werden. Zwar ist der Vorteil dieser Mechanismen intuitiv verständlich, aber Untersuchungen stehen bislang noch aus. So dürfte insbesondere für Unternehmen mit hohen gesetzlichen Bestimmungen interessant sein, wie es etwa bei Banken der Fall ist, welchen Mehrwert die Möglichkeit der Formulierung dynamischer SoD-*constraints* bietet.

Des weiteren wurde mit den Platzhalter-Attributen ein Prinzip vorgestellt, das mit sehr hoher Wahrscheinlichkeit ein große Zahl an Rollen unnötig macht, was darauf zurückzuführen ist, dass viele Zugriffsmuster in Unternehmen bis auf die konkreten Attributwerte identisch sind. An dieser Stelle sei nochmals das Beispiel aufgeführt, in dem zwei Bankangestellte das Recht haben, Kredite zu bewilligen. Ohne die Verwendung von Platzhaltern muss für jeden Kreditbewilligungsrahmen eine eigene Policy definiert werden, was in RBAC zur Entwicklung einer eigenen Rolle führt. Durch Platzhalter-Attribute würde dieser Sachverhalt in einer einzigen System-Policy formuliert werden können, da der Kreditbewilligungsrahmen im Benutzerobjekt selbst enthalten ist.

Ein weiterer Anknüpfungspunkt für Forschungsarbeiten stellt die Vererbung von Rechten dar, die erst durch die explizite Trennung von Geschäfts- und Systemrollen Vorteile aufweist. Es wurden mit *block policy inheritance* und *no override* zwei Prinzipien erwähnt, die zu einer Reduktion redundanter Policys einerseits und der Sicherstellung von Policy-Vorgaben andererseits führen. Dies könnte in einem BRBAC-konformen Rollenmanagementwerkzeug Beachtung findet. Es stellt sich demnach die Frage, ob zusätzliche Steuerungsmechanismen bei der Rechtevererbung als sinnvoll erachtet werden können.

Abschließend bietet BRBAC eine breite Palette an Möglichkeiten für Performance-Analysen an, um die Leistung dieses Rollenmodells unter verschiedenen Voraussetzungen zu messen. Die Erfahrungswerte, die aus diesen Tätigkeiten entstehen, könnten auch dazu führen, diese in Form von Erweiterungen in das BRBAC-Modell zurückfließen zu lassen.

Das Vorgehensmodell unterstützt den Entwurf der beiden Rollentypen aus BRBAC durch den hybriden Vorgehensansatz. Es stellt sich an dieser Stelle die Frage, ob dieser Ansatz im industriellen Einsatz als angemessen erachtet werden kann, da die Anzahl der Geschäftsrollen in etwa um zwei Größenordnungen kleiner ist, als die Anzahl an Systemrollen. Zwar wird die Entwicklung von Systemrollen durch die Anwendung von *role-mining*-Algorithmen unterstützt, was die Entwicklungsdauer tendenziell verringert, jedoch bleibt zu untersuchen, ob dieser Ansatz auch für große Einrichtungen praktikabel ist. Zwar ist dieses Vorgehen intuitiv verständlich und erscheint sinnvoll, aber Erfahrungswerte aus dem praktischen Einsatz, die die Angemessenheit des Vorgehens belegen, sind noch zu sammeln.

In der Implementierungsphase des Vorgehensmodell wird eine schrittweise Implementierung des Rollenmodells im Parallelbetrieb empfohlen, um Erfahrungswerte direkt in das Rollenmodell zurückführen zu können. Es wurde auch bereits erwähnt, dass dieses Vorgehen gerade in großen Einrichtungen ressourcenintensiv ist – sowohl auf personeller, als auch auf technischer Ebene. Zweifelsohne führt dieses Vorgehen zu einem passgenauen Rollenmodell für ein Unternehmen, jedoch muss etwa durch prototypische Projektumsetzungen gezeigt werden, wie ressourcenintensiv dieses Vorgehen tatsächlich ist, zumal die Passgenauigkeit des Rollenmodells durch die intensive Einbindung des Unternehmens in den Entwicklungsprozess bereits unterstützt wird.

Rollenmanagementwerkzeuge verkörpern integrative Zugriffskontrollarchitekturen. Aus der Betrachtung bestehender Werkzeuge in Kapitel 6 geht hervor, dass beide Werkzeuge keine native Unterstützung dienstorientierter Architekturen anbietet. Einerseits wird die Kommunikation der Werkzeuge mit den technischen Endsystemen entweder direkt oder unter Zuhilfenahme eines Provisionierungswerkzeug ermöglicht und andererseits kann keines der Werkzeuge nativ als Dienstgeber für parallele Zugriffskontrollsysteme dienen. Ein Ansatz, der sich in der Zukunft durchsetzen könnte, ist, dass jedes der technischen Endsysteme für die Integration mit Zugriffskontrollarchitekturen über Standardschnittstellen abgefragt werden kann und die das Zugriffskontrollsystem selbst ebenfalls in einem Dienstgeber/Dienstnehmer-Verhältnis auftreten kann. Das Vorhandensein einer Standardschnittstelle, etwa in Form von Webservice-Schnittstellen, würde die Verbreitung integrativer Architekturen für die Zugriffskontrolle stark vorantreiben. An dieser Stelle sei verwiesen auf [Em08] und [Di08], die sich mit der Thematik der Dienstorientierung von Zugriffskontrollarchitekturen genauer befassen.

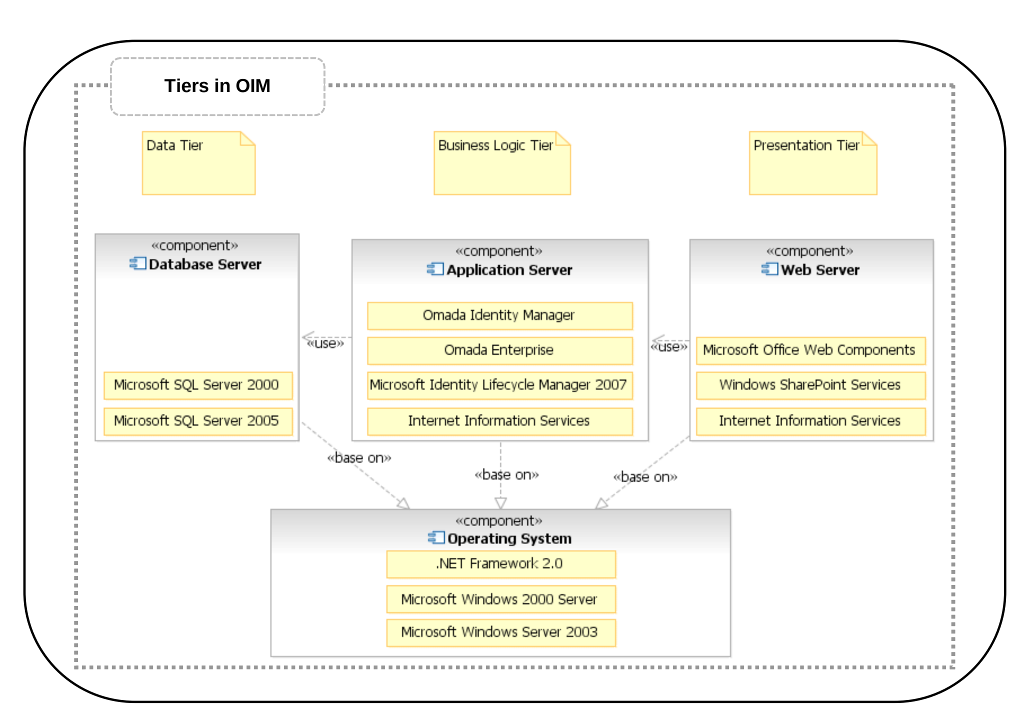
ANHANG

A Installation und Konfiguration des Omada Identity Manager

In diesem Kapitel wird der Prozess zur Installation und Konfiguration des Omada Identity Manager (OIM) besprochen, der in Kapitel 3.2 detailliert vorgestellt und in Kapitel 6.2 anhand eines Kriterienkatalogs bewertet wurde. Die Informationen hierfür sind den Installationshandbüchern des Omada Identity Manager und Omada Enterprise entnommen [OIM07a, OIM07b, OIM08]. Dabei wird zunächst die Installation der Basissysteme besprochen und anschließend die Installation und Konfiguration des Omada Identity Manager, die sich in die beiden Bereiche Omada Enterprise und Omada Identity Manager unterteilt.

A.1 Installation der Basissysteme

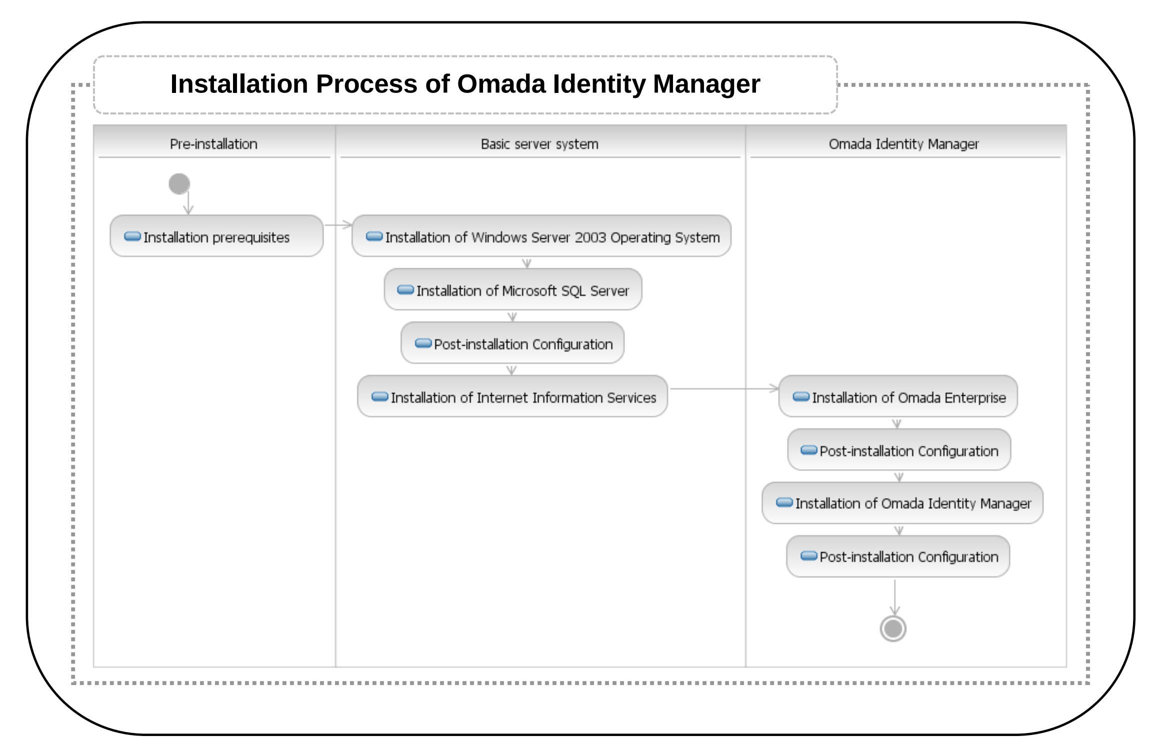
Der Identity Manager (OIM) ist ein kommerzielles Produkt der Firma Omada zur Verwaltung von Rollen. Es basiert auf Omada Enterprise (OE), einer Intranetz-Lösung zur Identitätsverwaltung. Der Identity Manager erweitert den Funktionsumfang von OE speziell um die Fähigkeit zur Verwaltung von Rollen. OIM ist eine klassische 3-Schichten-Architektur (engl. *three tier architecture*), bestehend aus Präsentationsschicht (engl. *presentation tier*), Logikschicht (engl. *business logic tier*) und Datenhaltungsschicht (engl. *data tier*) und basiert auf einer reinen Microsoft Serverumgebung. Jede dieser Schichten benötigt Microsoft Windows Server als Basisbetriebssystem. Dabei können die Versionen Microsoft 2000 Server oder Microsoft Server 2003 verwendet werden. Für die Datenhaltung unterstützt OIM Microsoft SQL 2000 und Microsoft SQL 2005 als relationalen Datenbankserver. In der Logikschicht werden die Internet Information Services (IIS) mit dem .NET Framework in der Version 2 für ASP.NET 2.0 benötigt, um darauf Omada Enterprise und Omada Identity Manager zu betreiben. Da die Funktionen des .NET Framework auch in den anderen beiden Schichten benötigt werden und im Zuge der automatischen Updates des Betriebssystems automatisch mitinstalliert werden, ist diese Komponente in Information 59 in der Komponente „Betriebssystem“ (engl. *operating system*) dargestellt. Der Zugriff auf die Datenbestände der einzelnen Endsysteme geschieht in Form des Microsoft Identity Lifecycle Manager (ILM), der ebenfalls auf der Logikebene vorhanden sein muss.



Information 59: Appendix – Tiers in Omada Identity Manager

Information 59 zeigt die drei Schichten in OIM und die Aufteilung der Komponenten darin. Alle drei Schichten können auf unterschiedlichen Rechnern betrieben werden, was aus Skalierbar-

keitsgründen durchaus sinnvoll ist. Wie bereits erwähnt wurde, basieren alle drei Schichten auf einem Microsoft Betriebssystem mit einer Installation des .NET Framework in der Version 2.0. Als Datenbankserver kommt entweder Microsoft SQL Server 2000 oder Microsoft SQL Server 2005 zum Einsatz. Dieser wird von OIM für dessen interne Datenhaltung benötigt und von ILM für die Speicherung dessen Verbindungsdaten zu den Endsystemen einerseits sowie der Managementagenten andererseits. Diese konkretisieren die Kommunikationsbeziehung zu den Endsystemen und spezifizieren, welche Daten von dort bezogen werden sollen. Zwar ist OIM lediglich eine Erweiterung zu Omada Enterprise, allerdings wird im Folgenden zur Vereinheitlichung lediglich von OIM gesprochen und nicht explizit unterschieden, welche Funktion OE bereitstellt und welche durch OIM hinzukommen, da dies durch die fehlende Dokumentation teilweise nicht möglich ist [OIM07a]. Im Folgenden wird nun auf die Installation des OIM eingegangen.



Information 60: Appendix – Installation Process of Omada Identity Manager

Die Installation des Omada Identity Manager ist in Information 60 als Aktivitätsdiagramm dargestellt. Hierbei wird implizit davon ausgegangen, dass die Gesamtarchitektur von OIM auf einer einzigen Maschine betrieben wird. Dies ist allerdings keine starke Einschränkung, da die Schichten von OIM aufgrund dessen Software-Architektur auf mehrere Systeme verteilt werden können. Zur Einhaltung eines gewissen Sicherheitsstandards sollte die Präsenationsschicht, auf der Benutzer mit dem System interagieren, von der Geschäftslogik und der Datenhaltung losgelöst sein. Um eine möglichst hohe Skalierbarkeit der Lösung zu gewährleisten, sollten alle drei Schichten auf dedizierte Computersysteme verteilt werden. Dieses Szenario verwendet jedoch der Einfachheit halber nur ein einzelnes Computersysteme. In einer realen Umgebung muss die Aufteilung der Komponenten auf die zur Verfügung stehenden Rechner allerdings wohlüberlegt sein. Dies ist in der Abbildung als „vorbereitende Maßnahmen vor der Installation" (engl. *pre-installation*) zusammengefasst. In dieser ersten Phase muss der Systemarchitekt somit die resultierende Architektur planen und die Systemanforderung von OIM analysieren. Als Artefakte entstehen dabei konkrete Installationspläne oder auch Migrationspläne, auf die hier nicht näher eingegangen wird.

Anschließend kann damit begonnen werden, das Basissystem zu installieren (engl. *basic server system*). In dieser Phase werden alle Komponenten installiert und konfiguriert, die die Installation von OIM voraussetzt. Nach der Installation des Windows Server-Betriebssystems wird der

Microsoft SQL Server als relationale Datenbank aufgesetzt. Sobald dies geschehen ist, wird das .NET Framework installiert, falls dies durch die automatischen Updates des Windows Servers nicht bereits durchgeführt wurde, weil die Omada Webanwendung speziell die ASP.NET Komponente dieses Rahmenwerk für den korrekten Aufbau der Benutzeroberfläche benötigt. Ferner werden die Office Web Components nachinstalliert, die ebenfalls auf der Oberfläche verwendet werden. Im Anschluss daran werden die Internet Information Services aufgesetzt, die Omada als Webserver voraussetzt. Sobald dies abgeschlossen ist, kann mit der eigentlichen Installation des Omada Identity Manager begonnen werden.

A.2 Installation und Konfiguration von Omada Enterprise

Wie bereits erwähnt wurde, muss zunächst Omada Enterprise als Serverkomponente installiert werden, ehe das Erweiterungspaket Omada Identity Manager aufgesetzt werden kann. OE stellt die grundlegende Funktionalität zur Identitätsverwaltung bereit, während der Omada Identity Manager diese durch eine Anpassung des Datenbankschemas um Rollenkomponenten erweitert. In diesem Zusammenhang erweitert OIM die bisherige Webanwendung um die Möglichkeit, Organisationseinheiten mit Hierarchien abzubilden sowie rollenbedingte Prozesse wie etwa das Delegieren von Arbeitsaufgaben an andere Rollen. Bei der Installation von OE wird das Installationsverzeichnis angegeben, welches nach der Installation als virtuelles Verzeichnis in den IIS eingebunden werden muss, um es über einen Web-Browser aufrufen zu können. Dazu wird über die Verwaltungskonsole des IIS eine neue Webanwendung hinzugefügt, mit dem Installationsverzeichnis von OE verknüpft und ASP.NET 2.0 als ausführendes Framework angegeben. An dieser Stelle kann auch der Authentifizierungsgrad für die Webanwendung angegeben werden. Hierbei wird unterschieden zwischen nicht-authentifiziertem Zugriff und der Authentifizierung gegen den zentralen Verzeichnisdienst. Zusammen mit OE wird die Provisionierungsplattform Microsoft Identity Lifecycle Manager (ILM) installiert, die für OIM die Kommunikation zu den Endsystemen realisiert. Die Verbindung zur Datenbank geschieht über ADO.NET und wird über die Configuration von Omada Enterprise vorgenommen. Hierbei sind der Datenbankserver, die konkrete Datenbank sowie der Sicherheitskontext in folgender Syntax anzugegeben:

```
          User ID=<Benutzername>; Password=<Passwort>; Initial
Catalog=<Datenbankname>; Data Source=<Datenbankserver>
```

Zu diesem Zeitpunkt ist Omada Enterprise über folgende Adresse erreichbar:

```
http://<Adresse_des_Servers>/<Name_der_Webanwendung>
```

Bei der Installation wurde ein Administrator in OE angelegt mit dem Benutzernamen „Administrator“ und einem leerem Passwort. Es wird empfohlen, zu diesem Zeitpunkt noch keine Systemkonfigurationen vorzunehmen, weil im nächsten Schritt die Erweiterung OIM installiert wird, die eine Schemaerweiterung der Datenbank vornimmt.

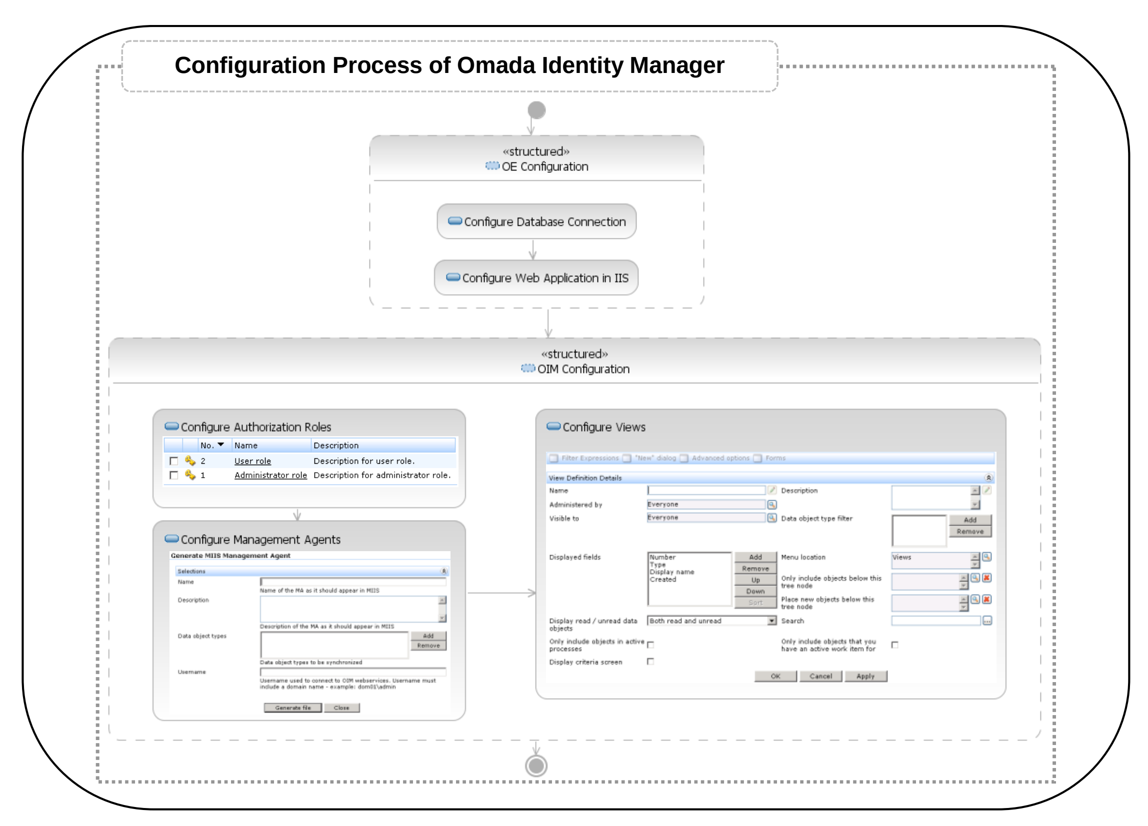
A.3 Installation und Konfiguration von Omada Identity Manager

Die Erweiterungen von OIM werden in das Verzeichnis von OE installiert und automatisch in IIS ergänzt. OIM verfügt über ein SQL-Script, welches in der Verwaltungskonsole des SQL-Servers ausgeführt werden muss. Dieses liegt in folgendem Ordner bereit:

```
<OIM-Installationsordner>\bin\scripts\
```

Die Ausführung des Scripts „dbcr-oim-X.sql“ erweitert das Datenbankschema und beendet die Installation des Omada Identity Manager. Nachdem die Installation von OIM abgeschlossen ist, kann dieser über die bereits erwähnte Adresse aufgerufen werden, um zusätzliche administrative Konfigurationen vorzunehmen. Beispielsweise kann ein Mailserver angegeben werden, der von OIM verwendet wird, um Benachrichtigungen etwa bei Fristüberschreitungen von Prozessindikatoren zu verschicken. Eine weitere grundlegende administrative Aufgabe ist das Anlegen von

Benutzerkonten, die auf OIM zugreifen. Dies lässt sich in OIM mit dem zentralen Verzeichnisdienst integrieren, so dass die Benutzerkonten automatisch in OIM importiert werden. Auch ist es innerhalb der Spezifikation von Geschäftsprozessen möglich, dass automatische Emails verschickt werden. Eine weitere administrative Aufgabe ist das Erstellen von Managementagenten für den ILM. Für ein tiefgreifenderes Verständnis dieser Komponenten sei an dieser Stelle verwiesen auf [OIM07b]. Ein drittes Beispiel einer administrativen Aufgabe ist das Definieren der rollenspezifischen Ansichten (engl. *views*). Der gesamte Konfigurationsprozess ist abschließend dargestellt in Information 61.



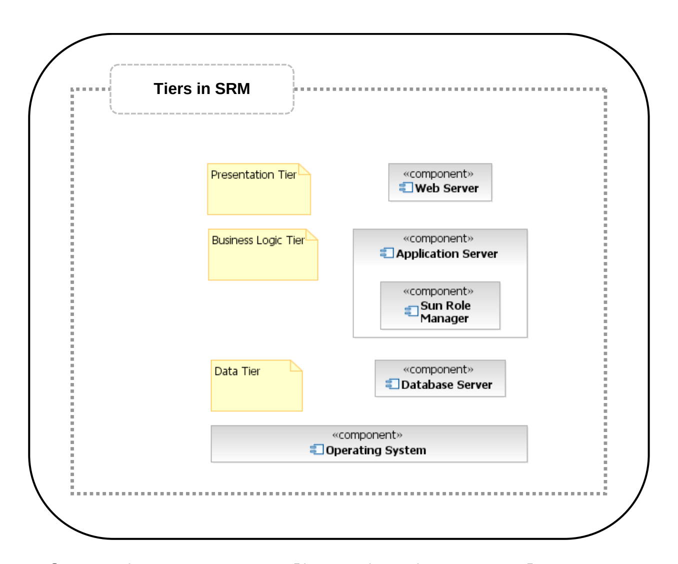
Information 61: Appendix – Configuration Process of Omada Identity Manager

B Installation und Konfiguration des Sun Role Manager

Dieses Kapitel beschreibt die Installation und Konfiguration des Sun Role Manager (SRM). Dieses Rollenmanagementwerkzeug wurde in Kapitel 3.3 detailliert beschrieben und in Kapitel 6.3 anhand eines Kriterienkatalogs bewertet. Die Informationen stammen dabei aus dem Installationshandbuch und dem Benutzerhandbuch [Vaau07a, Vaau07b].

B.1 Installation

SRM ist eine J2EE-Anwendung und benötigt neben einem Datenbankserver für die Datenhaltung einen Anwendungsserver für die Fachfunktionalität der Anwendung sowie einen Webserver um diese bedienbar zu machen. Diese drei Schichten sind in Information 62 veranschaulicht.



Information 62: Appendix – Tiers in Sun Role Manager

Im Folgenden wird auf die drei Schichten eingegangen. SRM unterstütz hierbei eine Vielzahl unterschiedlicher Datenbank-, Anwendungs- und Webserver. In der vorliegenden Version 4.0 unterstützt SRM folgende Datenbankserver:

- Microsoft SQL Server 2000(SP4)/2005
- IBM DB2 8.2, 9.x
- Oracle 9i, 10g, 11.x

Die unterstützten Anwendungsserver sind:

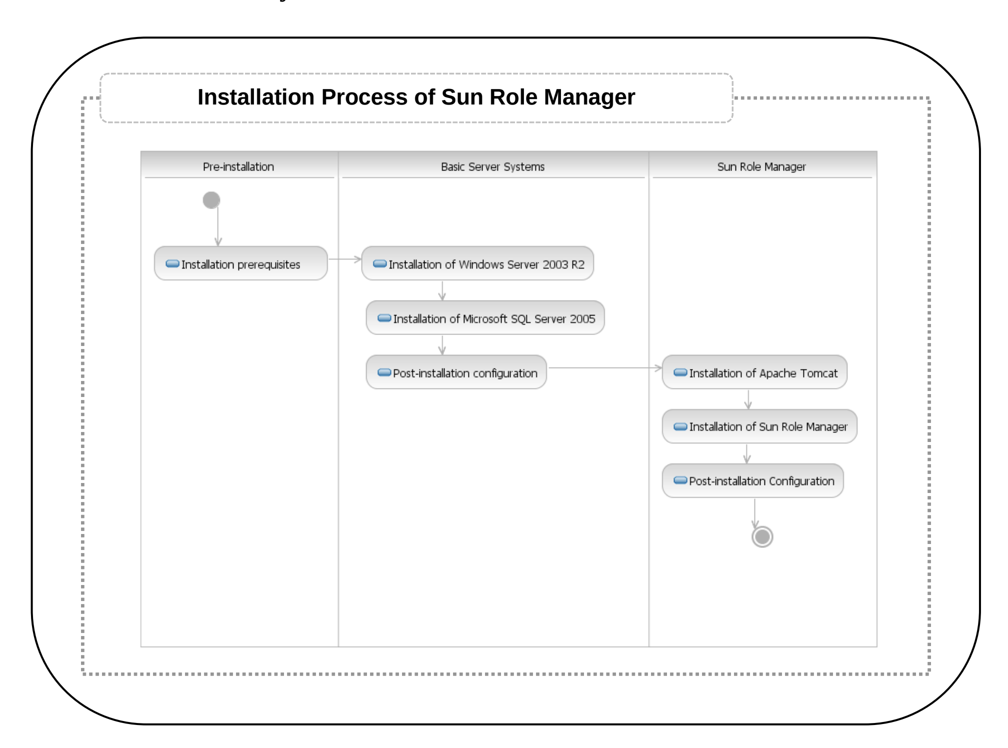
- Apache Tomcat 5.5
- IBM WebSphere 5.x, 6.x
- Weblogic
- JBoss
- Sun Java Application Server

Die unterstützten Betriebssysteme sind:

- Microsoft Windows Server 2000(SP3)
- Microsoft Windows Server 2003
- Solaris 8,9,10
- Red Hat Linux 4, 5
- Novel SuSE Linux Enterprise 9, 10

Die aktuelle Version des Sun Role Manager beinhaltet Apache Tomcat 5.5.16, welcher sowohl als Anwendungs- als auch als Webserver dient. Diese Arbeit verwendet die Variante mit dem Apache Tomcat 5.5.16 als Anwendungs- und Webserver und Microsoft SQL Server 2005 als

Datenbankserver. Als Basisbetriebssystem wird Microsoft Windows Server 2003 R2 eingesetzt. Im Folgenden wird nun zunächst die Installation des Apache Tomcat sowie des Microsoft SQL Servers demonstriert. Anschließend wird SRM installiert und für den Einsatz des Szenarios in Kapitel 7 dieser Arbeit konfiguriert. Es wird von einer bestehenden Installation des Windows Servers 2003 R2 ausgegangen, auf die hier nicht eingegangen wird. Eine schrittweise Anleitung zur Installation des Betriebssystems findet sich unter [Mic05a].



Information 63: Appendix – Installation Process of Sun Role Manager

In Information 63 wird der Prozess zur Installation der Werkzeugumgebung als Aktivitätsdiagramm dargestellt. In der Vorabphase (engl. *pre-installation*) wird damit begonnen, die Systemanforderungen zu analysieren. Dies umfasst die Systemanforderungen für die Systeme, die SRM zugrunde liegen und des Sun Role Manager selbst. Neben der Analyse der Systemanforderungen muss eine geeignete Netzwerkstruktur für die eingesetzten Systeme definiert werden. Da SRM auf einem 3-Schichten Modell aufbaut ist es möglich, dass die benötigten Server auf unterschiedliche Rechner verteilt werden und die Kommunikation zwischen den Serversystemen über das Netzwerk stattfindet. Dies gewährleistet eine Skalierbarkeit des Gesamtsystems, da die Last über das Netzwerk auf mehrere Rechner verteilt wird. In dieser Arbeit wird der Einfachheit halber nur ein einzelner Rechner für alle drei Schichten der Software-Architektur verwendet. Diese Entwurfsentscheidung ist allerdings für den produktiven Einsatz nicht zu empfehlen und wurde lediglich zur Vereinfachung des Installationsprozesses gefällt.

Nach den grundlegenden Entwurfsentscheidungen zum Systementwurf wird bei der Installation der Basissysteme (engl. *basic server systems*) damit begonnen, ein Betriebssystem zu installieren. In dieser Arbeit wird Windows Server 2003 R2, Enterprise Edition verwendet. Die Installation läuft menügetrieben und hält sich an das empfohlenen Standardvorgehen aus [Mic05a]. Im Anschluss an die Installation des Betriebssystems wird der Datenbankserver installiert. In dieser Arbeit wird der Microsoft SQL Server 2005 verwendet. Zur Installation sind administrative Rechte auf dem Betriebssystem nötig. Das Verzeichnis, in dem der SRM installiert werden soll muss vor dem Starten der Installation bereits angelegt werden, da es sonst während der Installation Fehler auftreten. Die Installationsprozedur des SRM wird durch eine grafische Oberfläche unterstützt und erfasst neben dem Installationsverzeichnis die Verbindung zum Datenbankserver sowie dem Anwendungsserver. Damit SRM im späteren Betrieb mit der Datenbank kommunizieren kann, wird ein für die Datenbank spezifischer Treiber benötigt. Da bereits bei der Instal-

lation des Role Manager eine Verbindung zum Datenbankserver hergestellt wird, um dort die vom SRM verwendete Datenbank anzulegen, muss bereits vor der Installation der hierfür benötigte JDBC-Treiber vorhanden und der Installationsprozedur mitgeteilt werden. Daher wird der Klassenpfad für die Installationsprozedur mit folgendem Kommandozeilenbefehl um den Pfad des Treibers erweitert:

```
set CLASSPATH=.;<$SRM_DRIVER>\<file name>
```

Da Apache Tomcat Teil des Installationspakets ist, erfordert der Anwendungs- und Webserver keine separate Installation, sondern geschieht zusammen mit SRM selbst. Es muss an dieser Stelle lediglich die Verbindungskennung (engl. *port*) angegeben werden, unter der der Webserver erreichbar sein soll. Zum Abschluss der Installation ist es möglich, ein automatisiertes Installationsscript exportieren zu lassen, um SRM mit dieser Konfiguration auf einem anderen Rechner zu installieren, oder die Installation erst zu einem späteren Zeitpunkt zu beginnen. Der Installationsprozess endet mit der Generierung eines Datenbank-Scripts, das das Schema, welches von SRM verwendet wird, im SQL-Server anlegt. Damit ist die Installation des Rollenmanagementwerkzeugs Sun Role Manager zusammen mit den Serversystemen abgeschlossen, auf denen es basiert.

B.2 Konfiguration

Am Ende der Installationsphase wurde ein Datenbank-Script generiert, welches in der Serverkonsole des SQL-Servers nun manuell ausgeführt werden muss. Innerhalb dieses Codestücks ist der Name der Datenbank sowie der für den Zugriff zu verwendende Benutzer bereits explizit aufgeführt. Dabei ist zu beachten, dass der Sun Role Manager in der vorliegenden Version nach einer Datenbank mit dem Namen „rbacx“ und einem Benutzerkonto „rbacxservice“ als Besitzer der Datenbank verlangt. Die Datenbank wurde bereits bei der Installation angelegt, der Benutzer muss jedoch manuell angelegt werden, ehe die weiteren Konfigurationsschritte initiiert werden können. Das Datenbank-Script wurde in folgendem Verzeichnis abgelegt:

```
<SRM-Installationsordner>\db
```

Die Ausführung des Scripts geschieht direkt im SQL Server, indem das Script zunächst über die Konsole geöffnet und anschließend die Datenbank „rbacx“ selektiert wird, auf die das Schema-Script angewandt werden soll. Am Ende dieser Konfiguration existiert die Datenbank zusammen mit dem von SRM benötigten Datenbank-Schema.

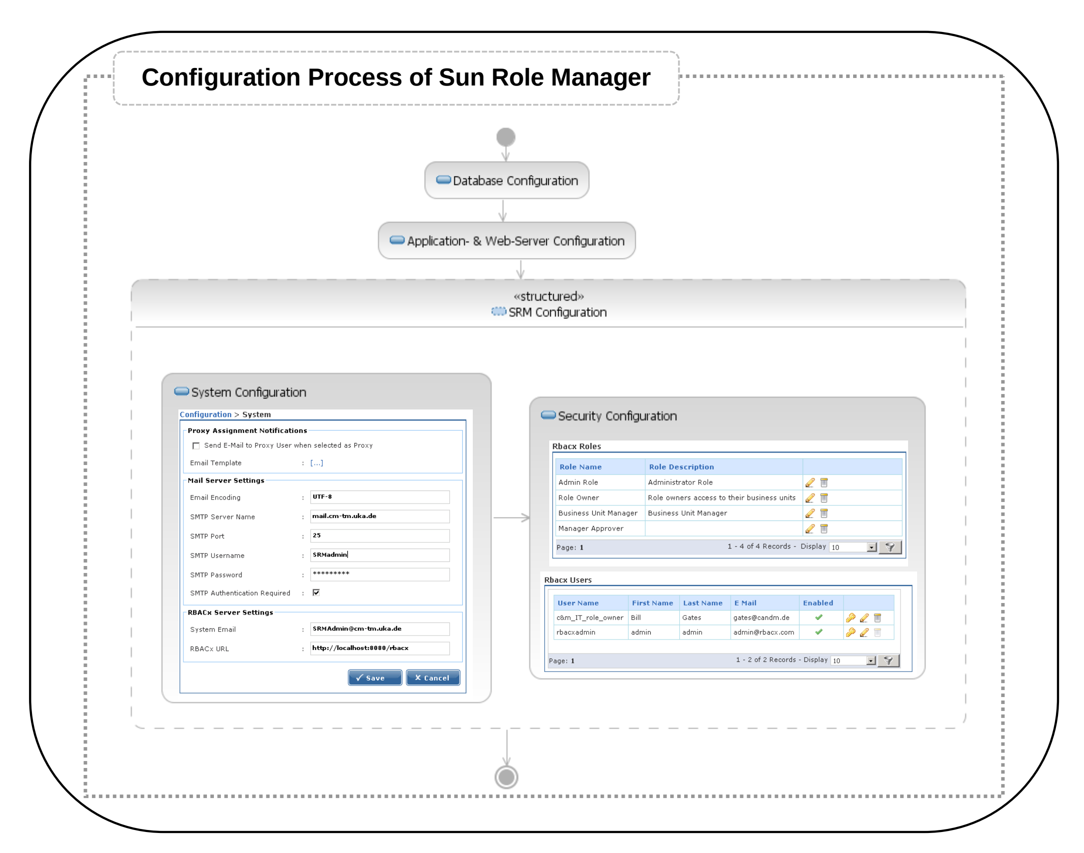
Nach der Konfiguration der Datenbank muss nun noch der Anwendungs- und Webserver angepasst werden. Apache Tomcat integriert sich als Windows-Dienst in das Betriebssystem und ist standardmäßig nicht gestartet. Der Sicherheitskontext, unter dem der Dienst läuft, ist standardmäßig auf „Lokales System“ gesetzt. Dadurch verfügt er über administrative Rechte auf dem Zielsystem. Aus Sicherheitsgründen wird empfohlen, hierfür ein dediziertes Benutzerkonto mit eingeschränkten Rechten zu erstellen. Um Tomcat und somit auch die darauf installierte Anwendung SRM, zu starten, muss nun lediglich der Dienst in der Dienst-Konsole des Windows Servers gestartet werden. Die Bedienoberfläche des SRM erreicht man ab diesem Zeitpunkt über:

```
http://<Adresse_des_Servers>:<Port>/rbacx
```

Laut [Vaau07a] liegt der empfohlene Speicherbedarf für den Role Manager bei 512 MB, was durch folgenden Kommandozeilenbefehl gesetzt und im Falle von Leistungsengpässen angepasst werden kann:

```
Set JAVA_OPTS="-Xmx512mb -Xms512m"
```

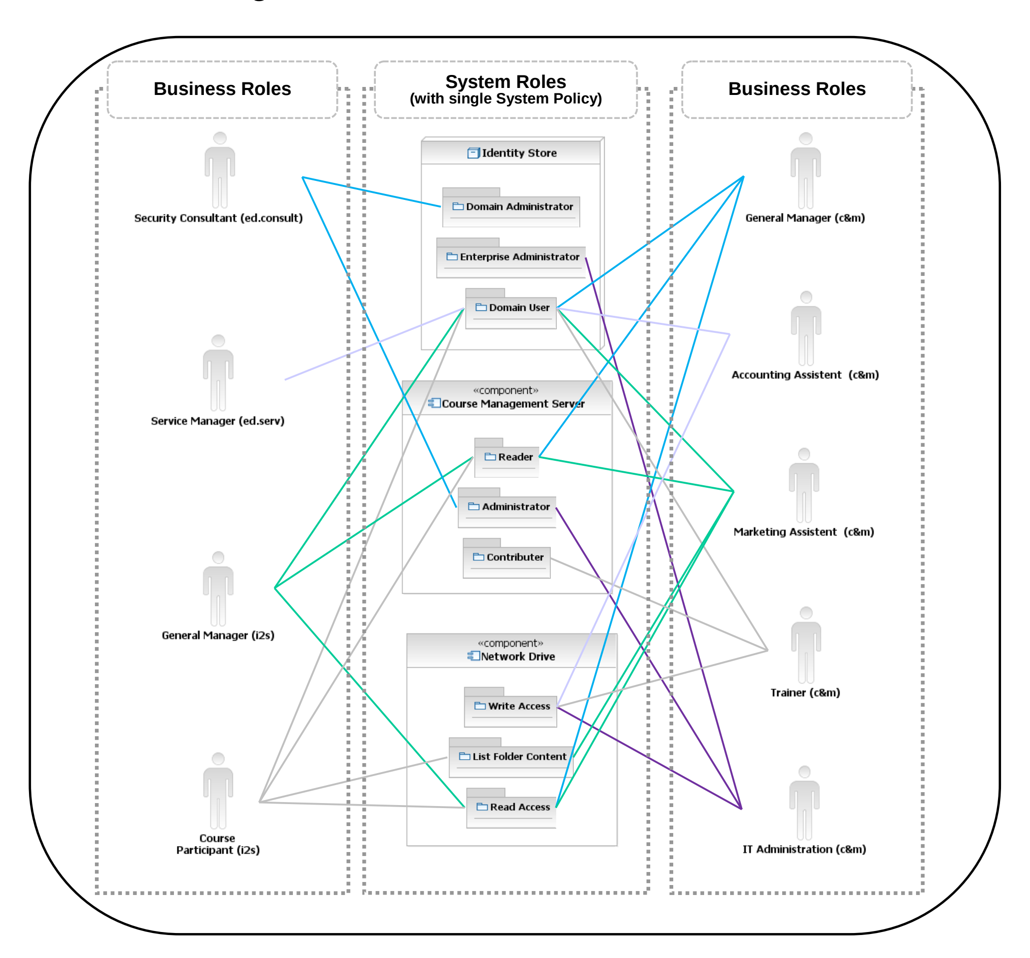
Nach der Konfiguration der Verbindungen zu Datenbank-, Anwendungs- und Webserver kann nun die Bedienoberfläche des SRM konfiguriert werden. Im Rahmen der Installation wurde ein administrativer Benutzer „rbacxadmin“ mit dem initialen Passwort „password“ angelegt. Das Passwort sollte unmittelbar geändert werden. Um anderen Benutzern den Zugriff auf SRM zu ermöglichen, können über die Bedienoberfläche Benutzer manuell angelegt und in SRM-eigene Rollen eingeteilt werden. Diese Rollen dienen lediglich der Steuerung des Rechteumfangs innerhalb von SRM. Neben der Konfiguration von Benutzern kann ein Mailserver spezifiziert werden, der zur Zusendung von Benutzerstatistiken verwendet werden kann. Diese beiden Konfigurationsschritte sind in Information 64 dargestellt. Damit endet die Konfiguration des SRM, der nun Einsatzbereit ist.



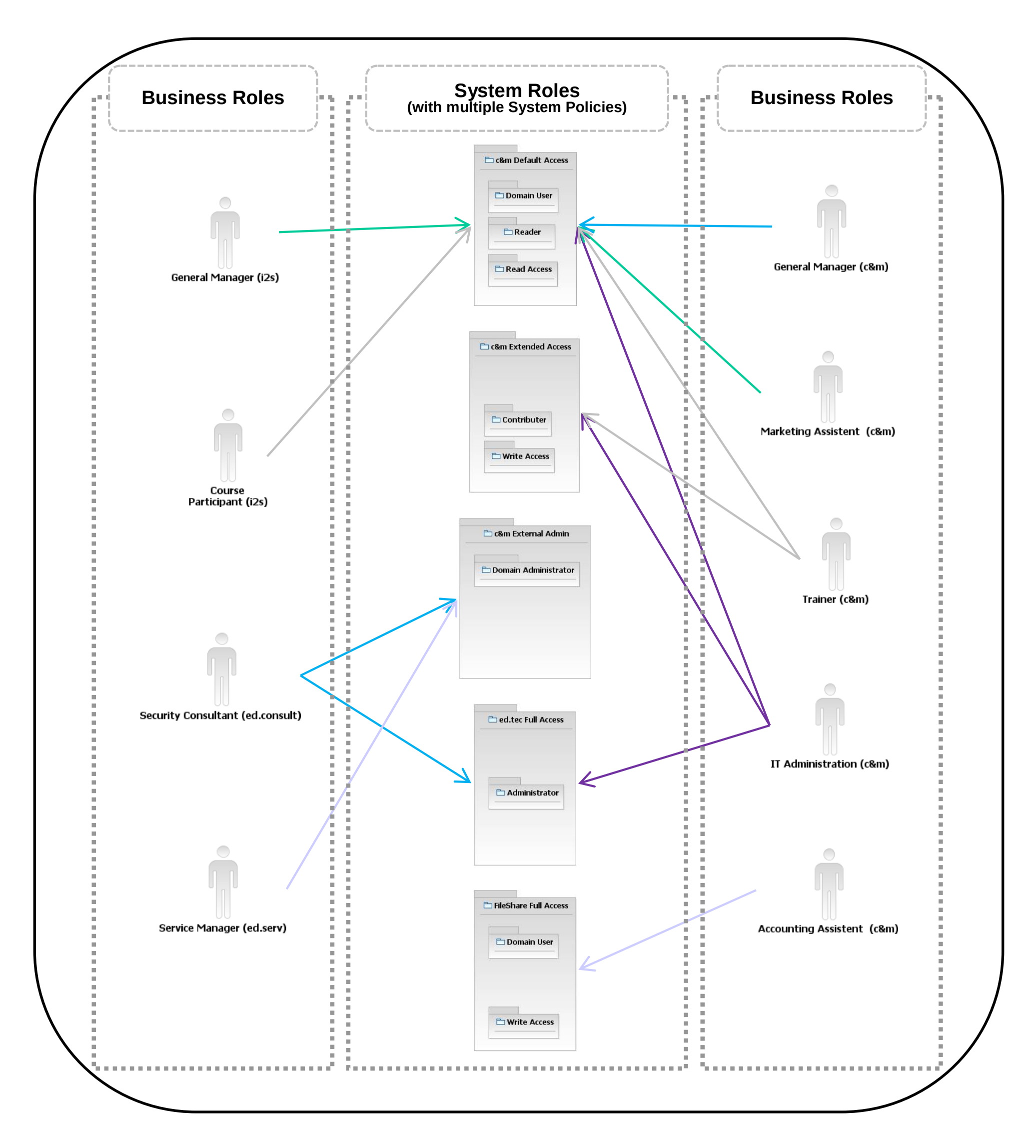
Information 64: Appendix – Configuration Process of Sun Role Manager

C Abbildungen aus dem IST-Szenario

Die beiden hier aufgeführten Abbildungen zeigen das Gesamtbild der Geschäftsrollen/Systemrollen-Relation aus Kapitel 7.2. Sie beziehen sich dabei auf Information 56 und Information 57. Dort wurde zum besseren Verständnis und für eine vereinfachte Darstellung lediglich ein Ausschnitt dargestellt.



Information 65: Appendix – Role Mapping with Simple System Roles



Information 66: Appendix – Role Mapping with Complex System Roles

D Abkürzungen

Abkürzung	**Langbezeichnung und/oder Begriffserklärung**
ACL	*Access control list* (dt. Zugriffskontrollliste)
ACM	*Association for Computing Machinery*
ADO.NET	*ActiveX Data Objects* auf .NET Technologie
ASP.NET	*Active Server Pages* auf .NET-Technologie
BR	*Business role* (dt. Geschäftsrolle)
BRBAC	*Business-focused role-based access control*
C&M	Cooperation and Management
c&m	*cooperation&more* Organisation innerhalb des IST-Szenarios
CMS	Content-Management-System
CSV	*Comma-separated values*
dSod	Dynamisches SoD
ed.consult	*education consulting* Organisation innerhalb des IST-Szenarios
ed.serv	*education services* Organisation innerhalb des IST-Szenarios
ed.soft	*education software* Organisation innerhalb des IST-Szenarios
ed.tec	*education technology* Schulungsportal aus dem IST-Szenario
ERBAC	*Enterprise role-based access control*
i2s	Organisation innerhalb des IST-Szenarios
IBAC	*Identity-based Access Control* (dt. identitätsbasierte Zugriffskontrolle)
ILM	Microsoft Identity Lifecycle Manager
IST	*Internet-supported training*-Szenario
IT	*Information technology*
J2EE	Siehe Java EE
Java EE	*Java Platform, Enterprise Edition* (früher: J2EE)
JDBC	*Java Database Connectivity*
KonTraG	Gesetzt zur Kontrolle und Transparenz im Unternehmensbereich

LDAP	*Lightweight directory access protocol*
MOSS	Microsoft Office SharePoint Server
NIST	*National Institute of Standards and Technology*
NIST-RBAC	Von NIST zertifiziertes RBAC-Standardrahmenwerk
OE	Omada Enterprise
OIM	Omada Identity Manager
OMG	*Object Management Group*
Op	*Operation*
PA	*Permission assignment*
PIP	*Policy Information Point*
RBAC	*Role-based access control* (dt. rollenbasierte Zugriffskontrolle)
RBAC96	*Role-based access control model 1996*
RBACx	Rollenmanagementwerkzeug der Firma Vaau Incorporated
RH	*Role hierarchy*
SoD	*Separation of Duty*
SOX	*Sarbanes-Oxley Act*
SP	Service Pack
SR	*System role* (dt. Systemrolle)
SRM	Sun Role Manager
sSod	Statisches SoD
TS	*Technical system*
UA	*User assignment*
UML	*Unified Modeling Language*

E Abbildungsverzeichnis

F Literaturanalysen

F.1 Advanced Features for Enterprise-Wide RBAC

[Ke02] Axel Kern: *Advanced Features for Enterprise-Wide Role-Based Access Control*, ACSAC, 2002

Inhalte

Was sind die zentralen Inhalte, die in der Arbeit behandelt werden?

(I1) Der Autor beschreibt das Konzept der *enterprise roles*, welches Rollen verkörpert, die sich über mehrere IT-Systeme erstrecken können. Dazu wird das ERBAC-Modell eingeführt.

(I2) Ferner wird beschrieben, wie sich das ERBAC-Modell auf das RBAC-Modell aus [FS+01] stützt und welche Komponenten es erweitert wird.

Defizite

Welche Defizite bestehender Arbeiten und Lösungen werden als Motivation der eigenen Lösungen genannt?

(D1) Unternehmen verlangen aufgrund wirtschaftlicher und technologischer Veränderungen nach immer mehr Flexibilität und Effizienz, um wettbewerbsfähig bleiben zu können. Dies trifft sowohl für die Geschäftsprozesse als auch für die darunterliegenden technischen Systeme zu. Ein Defizit stellen hierbei die Verfügbarkeit und Verlässlichkeit von Informationen in Systemen dar, da die zugrunde liegende Sicherheits-Architektur nicht mit der gleichen Geschwindigkeit wie die Informationstechnologie selbst gewachsen ist.

(D2) Die Komplexität und Kosten wachsen durch immer vielfältigere Anforderungen und Anwendungen. Der Autor nennt dafür viele Beispiele, unter Anderem, dass Anwendungen auf immer mehr Plattformen eingesetzt werden, die Organisationsstrukturen von Unternehmen durch Zukäufe immer komplexer werden und die Erteilung von spezifischen Zugriffsberechtigungen eines Benutzers sich immer schwieriger gestaltet. Auf Seiten der technischen Administration wird dadurch immer mehr Spezialwissen benötigt.

(D3) Die Erfahrung aus Industrieprojekten zeigt, dass das ERBAC-Modell selbst zu viele Rollen bedingt, wofür der Autor zwei hauptsächliche Gründe nennt. Vielen Faktoren wie etwa die Zugehörigkeit zu einer Organisationseinheit, Jobfunktion oder Ort im Sinne einer Zweigstelle bestimmen die individuellen Zugriffsrechte. Anstatt unterschiedlicher Rollenhierarchien müsste für jede Kombination eine eigene Rolle erstellt werden. Als zweiten Grund nennt der Autor den Bedarf an feingranularer Kontrolle auf Anwendungsebene, die pro Granularität eine eigene Rolle benötigen würde.

Prämissen

Welche Einschränkungen und Vorgaben werden hinsichtlich der eigenen Lösungen gemacht?

(P1) Anders als im klassischen RBAC-Modell verfügt das hier vorgestellte ERBAC-Modell nicht über den Mechanismus der *session*, um unterschiedliche Benutzerkontexte zu unterstützen. Dies ist aufgrund der systemübergreifenden Natur des hier eingeführten Rollenbegriffs nicht zu realisieren. Stattdessen werden die erteilten Berechtigungen an die beteiligten Endsysteme weitergereicht.

(P2) Durch das Fehlen des Benutzerkontextes ist dynamisches SoD nur durch einen Umweg zu realisieren. Falls das zugrundeliegende Endsystem nicht selbst über die dynamische Aufgabentrennung verfügt, wird der Benutzerkontext durch unterschiedliche Benutzerkonten pro Aufgabe realisiert.

Lösungen

Was sind die eigenen Lösungen?

(L1) Die Anzahl an resultierenden Rollen sowie die resultierende Rollenstruktur werden durch Parametrisierung wesentlich vereinfacht. Diese Parametrisierung ist in Form von Attributen und Regeln realisiert.

(L2) Das Benutzerobjekt verfügt im hier erweiterten ERBAC-Modell über allgemeine oder spezifische Benutzerattribute, die ihn genauer spezifizieren. Diese Informationen werden zur automatischen Einteilung in Rollen oder Benutzeradministration herangezogen.

(L3) Generische Rollen erlauben das Erteilen generischer Zugriffsrechte, falls mehrere Endsysteme über eine ähnliche Struktur von Gruppen und Rechten verfügen. Für die Menge dieser Systeme wird eine generische Berechtigung definiert und einer Rolle zugewiesen. Bei der Zuweisung dieser Rolle an einen Benutzer kann eine Teilmenge angegeben werden, was dazu führt, dass der Benutzer die generische Berechtigung nur auf dieser Teilmenge von Systemen erhält.

(L4) Jokerberechtigungen dienen dazu, um eine Rolleneinteilung in Endsystemen vorzunehmen. Der Autor nennt als Beispiel einen Verzeichnisdienst mit mehreren Gruppen, deren Namen mit dem Präfix „ACCT“ beginnen, gefolgt von einer vierstelligen Zahl. Diese Zahl stellt eine organisatorische Einheit dar und existiert somit auch als Benutzerattribut. Für dieses Präfix wird eine Jokerberechtigung erstellt und einer Rolle zugewiesen. Zum Zeitpunkt der Zuweisung einer Rolle an einen Benutzer greift eine Regel, die das Benutzerattribut ausliest, es mit der Jokerberechtigung konkateniert und den Benutzer somit in die korrekte Gruppe im Zielsystem einfügt.

(L5) Da sich die Kompetenzen zweier Angestellter in der gleichen Geschäftsrolle nur in Befugnisgrenzen unterscheiden können, führt der Autor benutzerspezifische Beschränkungen ein. Als Beispiel verwendet er zwei Bankangestellte, die über unterschiedliche Bewilligungsgrenzen für Kredite verfügen. Die Beschränkungen greifen bei der Zuweisung von Berechtigung an Rolle sowie von Rolle an Benutzer und beziehen sich im Beispiel auf die Obergrenze zur Bewilligung von Krediten, was wiederum ein Benutzerattribut ist und somit durch Anwendung einer Regel überprüft werden kann.

(L5) SAM Jupiter implementiert. Basiert auf ERBAC. In Deployment Projekten Erfahrung gesammelt und daraus wiederum ERBAC Modell erweitert um Parametrisierung.

Nachweise

Welche Nachweise werden hinsichtlich der Tragfähigkeit der eigenen Lösungen geliefert?

(N1) Das ERBAC-Modell ist im kommerziellen Sicherheitsprodukt SAM Jupiter implementiert. Der Autor belegt durch Erfahrungen aus Industrieprojekten, dass die Umsetzung dieses Modells praktikabel ist.

(N2) Der Autor erwähnt, dass durch die Erweiterung des ERBAC-Modells um die Parametrisierung die Anzahl an Rollen nachweislich gesenkt werden kann.

Offene Fragen

Welche Fragen sind noch ungelöst geblieben bzw. stellen sich dem Leser?

(O1) Gibt es einen Beleg, dass sich die Verringerung der Rollenzahl sowie die Vereinfachung der Rollenstruktur auch in der Praxis bewähren?

(O2) Welcher zeitliche sowie geldwerte Aufwand entsteht bei der Migration einer bestehenden Unternehmensstruktur auf SAM Jupiter?

(O3) In wie weit begünstigt ein durchdachter *role engineering* Prozess das ERBAC-Modell?

(O4) Wie steht dieser Ansatz zum hybriden Vorgehen bei Aufspühren von Rollen? Wo gibt es Anknüpfungspunkte?

F.2 Vorgehensmodelle für RBAC in heterogenen Systemlandschaften

[WW07] Felix Wortmann, Robert Winter: *Vorgehensmodelle für die rollenbasierte Autorisierung in heterogenen Systemlandschaften*, Wirtschaftsinformatik 49, 2007

Inhalte

Was sind die zentralen Inhalte, die in der Arbeit behandelt werden?

(I1) Ziel der Autoren ist es, Vorgehensmodelle zur Integration der Autorisierung in heterogenen Systemlandschaften sowohl in der Forschung als auch in der Praxis darzustellen.

(I2) Dabei werden zunächst Grundlagen zur Autorisierung erläutert und anschließend der aktuelle Stand der Integration in der Forschung aufgezeigt. Die Situation in der Praxis wird durch Fallstudien aus Großunternehmen belegt.

(I3) Die Betrachtung dieser beider Bereiche führt zu einer Weiterentwicklung eines Vorgehensmodells für die Definition und die Implementierung von systemübergreifenden Rollen.

Defizite

Welche Defizite bestehender Arbeiten und Lösungen werden als Motivation der eigenen Lösungen genannt?

(D1) Aktuelle Forschungsbeiträge verfügen über ausführliche theoretische Modelle wie Objekt- und Metamodelle. Für eine umfassende und verbesserte Methode zur Integration der Autorisierung muss auf Erkenntnisse aus der Theorie in Kombination mit Erkenntnissen aus der Praxis eingegangen werden.

Prämissen

Welche Einschränkungen und Vorgaben werden hinsichtlich der eigenen Lösungen gemacht?

(P1) Der ERBAC-Standard bildet die Basis des zu entwickelnden Vorgehensmodells

(P2) Im Mittelpunkt dieser Vorgehensweise steht die Implementierung systemübergreifender Rollen, im ERBAC-Standard als Enterprise-Rollen bezeichnet.

(P3) Es wurden Beiträge aus der Forschung ausgewählt, die sich im Hinblick auf die Entwicklung eines entsprechenden Vorgehensmodells verwerten lassen. Dadurch weisen sie erstens einen expliziten Bezug zur systemübergreifenden Autorisierung auf, zweitens enthalten oder thematisieren sie ein Vorgehensmodell, drittens erlauben sie bezüglich ihres Abstraktionsgrads eine hinreichend konkrete Diskussion und viertens sind sie umsetzungsorientiert oder bereits in der Praxis umgesetzt worden.

(P4) Die Autoren stellen in ihrer Arbeit Praxisprojekte in Form von Fallstudien vor und nehmen dies als Ausgangspunkt für die Ableitung eines Vorgehensmodells. Inhalt dieser Fallstudien sind Vorgehensmodelle bestehend aus mehreren Phasen und darin enthaltenen Aktivitäten.

Lösungen

Was sind die eigenen Lösungen?

(L1) Die Autoren konsolidieren Ansätze aus Forschung und Praxis zu einem daraus abgeleiteten Vorgehensmodell, welches zugleich transparent aus der Praxis als auch theoriebasiert sein will.

(L2) Hierzu wird, ausgehend von den Fallstudien, ein induziertes Modell erstellt, welches Aktivitäten beider Studien in Überdeckung bringt.

Nachweise

Welche Nachweise werden hinsichtlich der Tragfähigkeit der eigenen Lösungen geliefert?

(N1) Das Vorgehensmodell leitet sich von Projekten ab, die seit mehreren Jahren produktiv sind und auf andere Systeme ausgeweitet werden. Diese Projekte verfügen über eine Vielzahl unterschiedlicher Systeme, aus denen auf Basis systemübergreifender Rollen unterschiedliche Berechtigungen integriert wurden.

Offene Fragen

Welche Fragen sind noch ungelöst geblieben bzw. stellen sich dem Leser?

(O1) Die Handlungsanweisungen in den Aktivitäten sind nicht spezifiziert.

(O2) Das Vorgehensmodell beleuchtet nur die Phasen bis zur Implementierung. Es existiert keine Anknüpfmöglichkeit für ein Prozessmodell, was Verwaltungsaufgaben oder Pflege beachtet.

(O3) Das Vorgehensmodell basiert sehr stark auf den Fallstudien aus den Unternehmen. Die Einbeziehung theoretischer Modelle wie ERBAC geht aus dem Vorgehensmodell nicht hervor.

(O4) Die Autoren nehmen ähnliche Ergebnisartefakte einzelner Phasen in den Fallstudien als Indikator dafür, diese Phasen in ihrem eigenen Vorgehensmodell in einer einzelnen Phase zusammenzufassen. Diese neue Phase besteht aus allen Aktivitäten aus den Fallstudien. Ist diese Annahme gütig, bzw. was ist der Vorteil dieser Zusammenfassung?

F.3 Role Mining – Revealing Business Roles using Data Mining Technology

[KS+03] Martin Kuhlmann, Dalia Shohat, Gerhard Schimpf: *Role Mining – Revealing Business Roles for Security Administration using Data Mining Technology*, ACM, 2003

Inhalte

Was sind die zentralen Inhalte, die in der Arbeit behandelt werden?

(I1) Die Autoren verwenden klassische *data mining*-Technologien zum Aufbau einer zentralen Rollendatenbank. Sie stellen einen Prozess zur Erkennung von Mustern in einer zentralen Zugriffsdatenbank vor der daraus sowohl Unternehmens- als auch technische Rollen ableitet.

(I2) Als Motivation dieses Ansatzes stellen sie Gemeinsamkeiten klassischer Technologien aus dem *data mining* Umfeld mit dem Identifizieren von Rollen heraus. Ziel des *data mining* stellt das Generieren von Wissen aus Informationen die ihrerseits aus Daten gewonnen werden dar. Dieses Prinzip lässt sich auch auf Unternehmen anwenden, wo unterschiedliche Systeme im Einsatz sind, die einen Ausschnitt der Gesamtzugriffsinformation eines Benutzers repräsentieren.

(I3) Die Autoren begründen die Auseinandersetzung mit dem Rollenmanagement damit, dass Unternehmen die Vorteile Kostenreduktion und *compliance* einer rollenbasierten Gesamtstruktur erkennen und davon profitieren wollen.

Defizite

Welche Defizite bestehender Arbeiten und Lösungen werden als Motivation der eigenen Lösungen genannt?

(D1) Die Vergabe von Zugriffsrechten und deren Kontrolle ist von zentraler Bedeutung. In Umgebungen, die nicht auf dem Rollenkonzept aufsetzen, senkt das die Produktivität und steigert die Kosten. Das Herausarbeiten von Rollen ist daher wichtig. Die Autoren sehen RBAC als Basis des Provisionierungsprozesses. Ein Defizit bei Unternehmen stellt die Scheu vor dem initialen Aufwand dar. Ein systematischer und werkzeugunterstützer Prozess muss gefunden werden.

(D2) Mit einer *top-down*-Vorgehensweise die Rollenfindung zu initiieren ist aufgrund der gewachsenen Strukturen wünschenswert, allerdings sehr schwierig. Die gewachsenen Strukturen sind zumindest teilweise inkorrekt und inkonsistent. Die Analyse von Benutzern in ähnlichen Geschäftsrollen liefert oftmals unterschiedliche Berechtigung. Der Ansatz der Autoren ist hier, *data mining* Technologien auf einen aggregierten Informationsstamm anzuwenden.

Prämissen

Welche Einschränkungen und Vorgaben werden hinsichtlich der eigenen Lösungen gemacht?

(P1) Das Vorgehen der Autoren basiert auf dem ERBAC Standard, welcher über keine Hierarchien, *constraints* und SoD verfügt.

(P2) Das Vorgehensmodell ist *bottom-up*, mit dem Ergebnis einer flachen Rollenstruktur, anstatt einer Hierarchie, aus Zugriffskontrollsystemen und Geschäftsrollen.

(P3) Der Ansatz der Autoren geht von einer zentralen Datenbank aus, die über Informationen aus allen im Einsatz befindlichen Anwendungssystemen mit eigener Zugriffskontrolle verfügt.

(P4) Die IT Abteilung muss im vorgestellten Prozess mit eingebunden sein, um das Resultat des *role mining* zu verifizieren. Dies geschieht vor der Implementierung.

(P5) Das Ausrollen der Rollenstruktur auf das Unternehmen geschieht Schritt-für-Schritt.

Lösungen

Was sind die eigenen Lösungen?

(L1) Das Vorgehensmodell der Autoren kann in drei Phasen eingeteilt werden, die in L2, L3 und L4 geschildert werden.

(L2) Um eine zentrale, unternehmensweite Informationsdatenbank zu erhalten, müssen systemspezifische Benutzer, Attribute und Gruppenmitgliedschaften aus unterschiedlichen Systemen identifiziert werden und diese dann in eine Gesamtdatenbank importiert werden. Ziel hiervon ist eine organisierte Migration mit zentraler Administration.

(L3) Auf diesen Informationsstamm werden die Methoden *association* und *clustering* des IBM Intelligent Minder for Data für statistische und semantische Informationen als Schlüssel zum Finden von Rollen angewandt. Dieser erlaubt das Auswählen von Datenportionen und das Iterieren mit unterschiedlichen statistischen Grenzwerten. Hieraus wachsen Rollendefinitionen. Assoziationen erzeugen Regeln, die angeben, wie oft Ereignisse in Objekten zusammen auftauchen mit dem Ziel, funktionale Rollen zu identifizieren. Clustering gruppiert Einträg mit ähnlichen Attributwerten zum Auffinden von Unternehmensrollen.

(L4) Auf das Ergebnis dieses iterativen Prozesses wird der SAM Role Miner zum Erzeugen eines Rollenschemas angewandt, welcher zwischen Geschäftsrollen und funktionalen Rollen unterscheidet. Der Begriff Rolle ist hierbei nicht zielsystemspezifisch. Als Administrationsplattform wird SAM eingesetzt. SAM ist ein systemübergreifendes Verwaltungswerkzeug zur automatisierten Datenbeschaffung und Rollenimplementierung. Die direkte Bindung von Rollen an Berechtigungen wird soweit möglich im Zielsystem vermieden, indem dort Gruppen angelegt werden, auf die Berechtigungen erteilt werden. Als letzten Schritt werden Benutzer zu Rollen zugewiesen.

(L5) Ein Vorteil dieser Vorgehensweise ist das Profitieren von bisherigen Daten. Die von den Autoren vorgestellte Lösung erlaubt das Überprüfen des Ergebnisses des *role mining* an der Gesamtstruktur und somit das schrittweise Verfeinern. Dieser iterative Prozess nutzt die Lernkurve beim *role mining* aus.

Nachweise

Welche Nachweise werden hinsichtlich der Tragfähigkeit der eigenen Lösungen geliefert?

(N1) Als Nachweis wird auf Industrieprojekte und die Implementierung des kommerziellen Produkts SAM Role Miner verwiesen und durch zwei Fallstudien belegt.

(N2) 930 SAM Modelle wurden in zeitintensiver Arbeit vorab identifiziert. Innerhalb weniger Stunden konnten alle 930 Modelle als Rollen identifiziert werden. Dadurch verringert sich die Arbeitslast um zwei Größenordnungen.

(N3) Die Autoren schätzen eine Ersparnis von bis zu 60 % für den initialen Aufwand zur Rollenerstellung und von bis zu 50 % im laufenden Betrieb.

Offene Fragen

Welche Fragen sind noch ungelöst geblieben bzw. stellen sich dem Leser?

(O1) Welcher Grad an Automatismus ließe sich mit *role mining* in Kombination mit einem *top-down*-Vorgehen bei Unternehmen mit guten Geschäftsprozessbeschreibungen erreichen?

(O2) Wie gut wäre eine Kombination aus *top-down*- und diesem hier vorgestellten *bottom-up*-Vorgehen?

(O3) Was würde es bringen, wenn man vom vollständigen RBAC Modell ausgeht mit Hierarchien statt flacher Rollenstrukturen? Die Autoren schlagen einen zusätzlichen *mining*-Schritt nach dem Identifizieren der Rollen zum Etablieren von Hierarchien vor.

(O4) Wie wirkt sich das entstehende Rollenmodell aus, wenn man nicht von Benutzer/Rolle-Paaren, sondern in einem vor gelagerten Schritt zunächst von Anwendung/Berechtigungs-Paaren ausgeht? Würde man hier Aggregatobjekte für Zugriffsrechte innerhalb von Anwendungen schaffen. Könnte dies dem SoD-Prinzip zweckdienlich sein?

(O5) Hilft eine Parametrisierung zur weiteren Verfeinerung des *role-mining* Prozesses?

(O6) Wie lassen sich Geschäfts-Policys besser mit einbeziehen, damit Hierarchien, SoD oder *constraints* unmittelbarer ersichtlich werden?

(O7) Wie kann die hier vorgestellte Lösung um statische *constraints*, SoD, Hierarchien und Parametrisierung kann ergänzt werden?

(O8) Es ist immer noch ein großer manueller Aufwand nötig, um den Cluster auf eine geschäftsartige Sicht hin zuzuschneiden. Deshalb werden mit diesem Ansatz weniger als 80 % Automatisierungsgrad bei der Rollenidentifikation erreicht.

(O9) Eine Voraussetzung ist, dass ein Benutzer pro Endsystem über maximal ein Benutzerkonto verfügt. Dies ist laut eigener Aussage eine realistische Annahme. Jedoch wird in den heute vorherrschenden identitätsbasierten Zugriffskontrollarchitekturen der unterschiedliche Rechteumfang eines Benutzers gerade durch Identitäten voneinander getrennt. Ein Benutzer hat somit bei der Ausführung unterschiedlicher Aufgaben heutzutage mehrere Benutzerkonten im selben Endsystem. Dies lässt sich im identitätsbasierten Paradigma nicht anders lösen.

F.4 Observations on the Role Life-Cycle

[KK+02] Axel Kern, Martin Kuhlmann: *Observations on the Role Life-Cycle in the Context of Enterprise Security Management*, Proceedings of the seventh ACM symposium on Access control models and technologies, Monterey, Kalifornien (USA), Seiten: 43 - 51, 03.-04. Juni 2002

Inhalte

Was sind die zentralen Inhalte, die in der Arbeit behandelt werden?

(I1) In dieser Arbeit wird ein Lebenszyklus von Rollen vorgestellt, angelehnt an den klassischen Software-Entwicklungsprozess.

(I2) Dabei wird untersucht, wie sich diese Zyklen in ein theoretisches Rahmenwerk eingefügt werden können, um es Neueinsteigern leichter zu machen, rollenbasierte Anwendungen zu entwickeln.

(I3) Die Autoren sind überzeugt, dass der Lebenszyklus einer Rolle als Basis für ein RBAC-Rahmenwerk angesehen werden kann. Sie liefern eine initiale Diskussion zu diesem Thema, basierend auf den Erfahrungen aus dem Sicherheitsmanagement.

(I4) Die Autoren stellen ein Lebenszyklus-Modell vor, das sich auf einen inkrementell-iterativen Prozess stützt, bestehend aus den Phasen Analyse, Entwurf, Verwaltung und Pflege.

Defizite

Welche Defizite bestehender Arbeiten und Lösungen werden als Motivation der eigenen Lösungen genannt?

(D1) Rollen als Paradigma im Sicherheitsmanagement etabliert sich bald als Standard und es gibt bereits einige Erweiterungen zu den Modellen, aber diese Ansätze fokussieren sich nur auf eine statische Sicht, mit dem Ziel, Rollen zu definieren. Der dynamische Ansatz eines Lebenszyklus wird nicht betrachtet.

(D2) In den letzten Jahren haben große Unternehmen einen Rollenbegriff geprägt, der auf die Unternehmensebene beschränkt ist. Diese Auffassung geht von der in D1 beschriebenen statischen Sichtweise aus, weil sich Rollen auf Unternehmensebene und damit einhergehend, geschäftliche Funktionen selten ändern. Sowohl die praktische Erfahrung der Autoren, als auch

jüngste Forschungsergebnisse zeigen jedoch, dass sich Rollen im Laufe der Zeit entwickeln, was den Ansatz von Prozessen ähnlich dem in der klassischen Software-Entwicklung rechtfertigt.

(D3) Die steigende Komplexität stellt ein weiteres Defizit gängiger Ansätze dar. Im Bereich allgemeiner rollenbasierter Zugriffskontrolle hat sich in den letzten Jahren sehr viel geändert. Der Aspekt eines Lebenszyklus von Rollen blieb dabei sowohl in der Theorie, als auch in der Praxis weitestgehend unbehelligt. Dies ist insbesondere in Anbetracht stets zunehmender Komplexität in Unternehmenssystemen ein Kriterium für den Ansatz der Autoren.

(D4) Die Autoren versprechen sich durch ihren Ansatz eine weitere Zunahme bei der Akzeptanz von RBAC-konformen Zugriffskontrollsystemen, weil der von ihnen propagierte Lebenszyklus eine natürliche Auffassung eines Lebenszyklus widerspiegelt. Er sorgt für ein besseres Verständnis des Rollenbegriffs als das bisherige Verständnis einer Rolle als eine „Mittelschicht“ zwischen Benutzern und Berechtigungen.

(D5) Die Motivation der Autoren, eine Diskussion über den Lebenszyklus von Rollen anzuregen, umfasst in erster Linie folgende vier Punkte: Der Lebenszyklus stellt Schnittstellen zur Einordnung existierender und zukünftiger Arbeiten im Bereich von Rollen dar. Zweitens bietet ein Lebenszyklus einen strukturierten Prozess zur Entwicklung eines rollenbasierten Gesamtsystems dar. Drittens stellt der Lebenszyklus eine effiziente Anwendung von Rollen in Anbetracht von Änderungen im Unternehmen dar und viertens ist ein Rollenlebenszyklus die Basis für einen geordneten Umgang mit Rollen (engl. *role engineering*).

Prämissen

Welche Einschränkungen und Vorgaben werden hinsichtlich der eigenen Lösungen gemacht?

(P1) Der Lebenszklus ist die Basis der Arbeit.

(P2) Die Motivation basiert auf den folgenden beiden Prämissen: Der Lebenszyklus von Rollen muss hinreichend abstrakt gehalten sein, um damit in unterschiedlichen Szenarien umgehen zu können, zumal eine Rolle ein kontext-abhängiges Konstrukt mit wechselnder Semantik ist. Zweitens muss man sich bewusst sein, dass man sich bei der Verwendung eines Rollenlebenszyklus Techniken zur Wiederverwendung, – bekannt aus der Software-Entwicklung – eingehen befasst hat.

Lösungen

Was sind die eigenen Lösungen?

(L1) Es wird ein *life-cycle*-Modell vorgestellt, basierend auf einem iterativen, aufeinander aufbauenden Prozess in Anlehnung an den klassischen Software-Entwicklungszyklus.

(L2) In der Analysephase (engl. *analysis*) wird die Organisationsform des Unternehmens und Rollen auf einer sehr abstrakten Ebene formalisiert und der Entwurfsphase (engl. *design*) in Form von Artefakten übergeben. Hier werden die Rollen technisch verfeinert und auf die Strukturen der Unternehmenssysteme transformiert. Sobald dies geschehen ist, beginnen in der Verwaltungsphase (engl. *management*) administrative Aufgaben. Da Rollen wie jede systematische Information Änderungen unterlegen ist muss in der Überprüfungsphase (engl. *maintenance*) sichergestellt werden, dass sich die Rollen sowie das implementierte Rollenmodell an diese Änderungen anpassen.

(L3) Die Autoren präsentieren ein Prozessmodell und weisen daraufhin, dass ein solches Modell immer nur eine gewisse Perspektive eines Entwicklungsprozesses darstellen kann. Auch in dieser Publikation wird nur ein Teilausschnitt vorgestellt.

(L4) Der klassische Software-Entwicklungsprozess, bestehend aus den Phasen Analyse, Entwurf, Implementierung, Test und Überprüfung wird ausführlich erklärt und über den iterativen Prozesscharakter der Bezug zum Lebenszyklus von Rollen hergestellt. Auch auf Nachteile bei der Übertragung des klassischen Software-Entwicklungsprozesses auf Rollen wird hingewiesen. Diese haben mit der fehlenden Rückkopplung zu tun und führen dazu, dass die Entwicklungszeit bis zum fertigen Produkt unter Umständen länger dauert als anvisiert. Dies hat laut den Autoren drei hauptsächliche Gründe: Der Ansatz spiegelt nicht die Situation auf dem freien Markt wieder, wo die Prämisse herrscht, Produkte schnell liefern zu können. Ferner wird oftmals erst beim Einsatz des fertigen Produktes festgestellt, dass vorab festgelegte Produkteigenschaften nicht abgedeckt sind. Drittens ändern sich Produktanforderungen im Laufe des Entwicklungsprozesses, so dass man von einem vollständigen Pflichtenheft nicht ausgehen kann. Moderne Software bedient sich daher eines iterativen, inkrementellen Entwicklungsprozess mit Rückkopplung, um früher in die Produktion zu gehen und Nachbesserungswünsche seitens des Kunden rechtzeitig entsprechen zu können.

(L5) Die Autoren setzen sich mit dem Begriff der *enterprise role* auseinander, einer Definition für systemübergreifende Rollen und führen ein Modell für unternehmensweite Rolle ein. In der Tat haben die unterschiedlichen Unternehmenssysteme meist ein eigenes Verständnis des Rollenbegriffs und durch die Unterschiedlichkeit der Technologien unterscheidet sich dieser Rollenbegriff teilweise sehr stark. Ein typischer Unternehmensbenutzer muss über Zugriffsrechte in unterschiedlichen Systemen verfügen, nur ist oftmals eine „Rolle“ ein auf ein einzelnes System eingeschränkter Begriff für Zugriffsrechte. Die Autoren halten es für unrealistisch, davon auszugehen, dass diese Rollenbegriffe eines Tages auf einen gemeinsamen Nenner gebracht werden können und ziehen den Ansatz von Unternehmensrollen als Behälter für die Menge an unterschiedlichen Auffassungen für „Rollen“ vor. Das von den Autoren vorgestellte Produkt „SAM“ dient daher nicht als Autorisierungsplattform, sondern als Plattform auf einer höheren Abstraktionsebene, die sich quer über alle Systeme erstreckt und die unterschiedlichen Systeme unter einer Plattform vereint.

(L6) Für die Implementierung von unternehmensweiten Rollen präsentieren die Autoren ein vierstufiges Prozessmodell an, bestehend aus der Konsolidierungsphase, in der unter anderem der Bestand an Rollen aufgenommen und analysiert wird (engl. *role analysis*). Gleichzeitig wird die Rollen-Software implementiert und die Unternehmenssysteme daran angeschlossen. Die zweite Phase befasst sich mit der Automatisierung, was den Entwurf von Automatismen innerhalb des Rollenentwurfs (engl. *role design*) betrifft. Parallel kann in dieser Phase die Verbindung zu den einzelnen Informationsbeständen der Unternehmenssysteme, wie etwa Datenbanken, hergestellt werden. Die nächste Phase des „Ausrollens“ befasst sich mit dem geplanten Einbetten der rollenbasierten Infrastruktur in die Gesamtlandschaft sowie dem Einsatz eventuell weiterer Werkzeuge, die in diesem Zusammenhang von nutzen sein könnten, wie etwa ein *Single-Sign-On*-System (SSO) oder ein *workflow*-System für die Rollenadministration. Die letzte Phase befasst sich mit den anschließend anfallenden Verwaltungsaufgaben, wie etwa der Pflege der Rollen (engl. *role management, maintenance*).

Nachweise

Welche Nachweise werden hinsichtlich der Tragfähigkeit der eigenen Lösungen geliefert?

(N1) Nachweise für den Lebenszyklus kommen aus Theorie und Praxis und werden von den Autoren ausführlich geschildert und durch Quellen belegt.

(N2) Als Nachweis aus der Praxis kommt der Security Administration Manager (SAM) zum Einsatz. SAM stellt ein unternehmensweites Benutzeradministrationswerkzeug dar, welches über einen zentralen Datenstamm für Benutzer- und Ressourcenverwaltungen verfügt. Allgemeines Ziel des SAM ist die zentrale Verwaltung mehrerer Systeme. Aufgrund der zunehmenden Komplexität von Unternehmenssystemen steigt auch der Bedarf an plattformspezifischem

Wissen dieser Systeme, um mit diesen in einem integrierten Ansatz kommunizieren zu können. Durch das Integrieren dieser Systeme in SAM verwendet die technische Administration nur eine einzelne Benutzeroberfläche und löst somit das Komplexitätsproblem in heterogenen Systemlandschaften. Neben den Kosten, die durch die Konsistenzhaltung unterschiedlicher Systeme entsteht, stellt die manuelle Administration einen zweiten Kostenfaktor dar, der durch SAM automatisiert werden kann. SAM bietet hierfür automatische Provisionierungs- und Deprovisionierungsprozesse zur Verfügung.

Offene Fragen

Welche Fragen sind noch ungelöst geblieben bzw. stellen sich dem Leser?

(O1) Der hier aufgezeigte Ansatz ist zwar nachvollziehbar und durch Praxiserfahrung belegt, jedoch an wesentlichen Stellen auf einfache Sachverhalte reduziert. So ist etwa das in dieser Publikation verwendete Rollemodell als Repräsentation der Organisationsstruktur sehr einfach gehalten. Wie verhält sich ein derartiger Ansatz jedoch in großen Unternehmen?

(O2) Innerhalb des Rollenzyklus sind die Phasen sehr klar voneinander zu trennen. Welche Artefakte jedoch entstehen innerhalb dieser Phasen?

(O3) Grundlegend wichtig an einem RBAC-Ansatz für heterogene Systemlandschaften ist, dass die Rollenmanagementlösung mit einer möglichst großen Zahl an Endsystemen kommunizieren kann. Darauf wird jedoch nicht eingegangen.

F.5 A role-based infrastructure management system

[SA+04] Dongwan Shin, Gail-Joon Ahn, Sangrae Cho, Seunghun Jin: *A role-based infrastructure management system: design and implementation*, John Wiley & Sons Ltd, 2004

Inhalte

Was sind die zentralen Inhalte, die in der Arbeit behandelt werden?

(I1) Die Autoren sehen einen deutlichen Zuwachs an RBAC-Aktivitäten in Theorie und Praxis. Sie sehen hierdurch insbesondere einen Vorteil für die IT-Verwaltung, da die Kopplung von Berechtigungen an Rollen fehlerresistenter und kostenreduktiver ist, als die Kopplung an einzelne Benutzer. Da die Verwaltung von Rollen in vielen Implementierungen auf einer ad-hoc-Basis geschieht, wird hier ein System vorgestellt, das gültige Rollen, Benutzer und Berechtigungen ermöglichen soll.

(I2) Das Produkt namens „RolePartner“ soll es Administratoren ermöglichen, gewisse Komponenten eines RBAC-Modells abzubilden, um Zugriffskontroll-Policys darzustellen. Somit bildet RolePartner eine solide Grundlage für den Betrieb einer rollenbasierten Infrastruktur. Dazu gehört das Etablieren von gültigen Rollen und Hierarchien samt Benutzern und Berechtigungen. Insbesondere müssen Rollen durch Beschränkungen (engl. constraints) wie etwa Zugriffskontroll-Policys klar voneinander abgegrenzt werden.

(I3) Dazu werden drei Methoden vorgestellt und Entwurfs- sowie Implemetierungsprobleme erörtert.

(I4) Zunächst folgt aber eine Einführung in den Bereich *role engineering* and anschließend in *role administration*. Dazu wird jeweils der Bezug zu aktuellen Forschungsarbeiten aufgezeigt.

(I5) *Role engineering*: Es wird ein Überblick über die drei Vorgehensarten im role engineering gegeben. Im *top-down*-Ansatz wird von Rollen ausgegangen und die dazu benötigten Systemberechtigungen verfeinert. Im *bottom-up*-Verfahren werden technische Zugriffsberechtigungen zu Berechtigungsgruppen zusammengefasst, um so Rollen und Rollenhierarchien zu etablieren. Der *middle-out*-Ansatz versucht, diese beiden Prinzipien zu verknüpfen.

(I6) *Role administration*: Zum Abschluss der Tätigkeiten des *role engineerings* steht ein Grundgerüst der RBAC-Umgebung mit Rollen oder anderen RBAC Komponenten. Hieran schließen sich nun Verwaltungsaufgaben an. Wie die Autoren schildern, lässt sich die Rollenadministration in die beiden Module Policy-Verwaltung (engl. *policy management*) und Policy-Anwendung (engl. *policy enforcement*) aufteilen, wobei sich *policy management* mit dem Vorhalten von Zugriffskontroll-Policys befasst und *policy enforcement* diese dann mit den RBAC-Komponenten verknüpft. Diese Aufgabe ist sehr gewissenhaft auszuführen, um die Richtlinien im Unternehmen korrekt in Policys abzubilden und nicht davon abzuweichen. Auch hier wird der Bezug zu aktuellen Forschungsbeiträgen geschaffen.

(I7) In der Publikation wird anhand eines LDAP-Verzeichnisses gezeigt, dass RolePartner nahtlos in eine bestehende Umgebung integriert werden kann, die nicht rollenbasiert arbeitet.

Defizite

Welche Defizite bestehender Arbeiten und Lösungen werden als Motivation der eigenen Lösungen genannt?

(D1) Viele Implementierungen ermöglichen Rollen lediglich auf einer ad-hoc-Basis.

(D2) Die drei Vorgehensweisen werden vorgestellt und die Schwächen daran an ausgewählten Beispielen anderer Autoren aufgezeigt. So sind Ansätze, die den *top-down*-Ansatz verfolgen klar in Modellen definiert, was den Ansatz wenig dynamisch erscheinen lässt. Beim *bottom-up*-Vorgehen wird Bezug genommen auf ein Werkzeug, dessen Schwäche es ist, dass es nur für Webanwendungen funktioniert. Im *middle-out*-Verfahren wird ein Beispiel gewählt, dass theoretisch sehr präzise formuliert ist, aber viele Fragen im Bezug auf die konkrete technische Umsetzung offen lässt.

(D3) Schwachstellen von aktuellen *role administration*-Ansätzen: Hierbei handelt es sich um einen Beitrag, der sich mit Rolle/Hierachie-Tupeln sowie Rolle/Rolle-Tupeln befasst und diese in einer RBAC-Datenbank vorhält. Die aufgezeigte Schwachstelle liegt hier daran, dass dieses System nur für Webanwendungen verwendet werden kann. Ein zweiter Beitrag befasst sich mit der Entwicklung flexibler RBAC-Dienste über die Sprache xoRBAC, vernachlässigt aber in den Augen der Autoren die Aspekte Rollen- und Policy-Verwaltung.

Prämissen

Welche Einschränkungen und Vorgaben werden hinsichtlich der eigenen Lösungen gemacht?

(P1) Ihre Lösung ist speziell für Unternehmen mit einer Vielzahl unterschiedlicher Unternehmenssysteme gedacht. Sie baut auf einer rollenbasierten Infrastruktur auf und schließt insbesondere die beiden Module *role engineering* und *role administration* mit ein.

(P2) Speziell die aus P1 resultierende große Zahl an Rollen macht eine Betrachtung von *role engineering* und *role administration* unausweichlich. In dieser Veröffentlichung liegt der Fokus klar auf dem Punkt *role administration*.

(P3) Die speziellen Anforderungen eines solchen Systems sind: Das Vorhandensein in möglichst vielen Unternehmenssystemen (engl. *availability*), die Anwendbarkeit in möglichst vielen Policy-Umgebungen, die auf RBAC aufbauen (engl. *applicability*), die Benutzerfreundlichkeit,

so dass ein Rollenadministrator intuitiv damit umgehen kann (engl. *ease-of-use*) und die Konformität zu RBAC-Standardrahmenwerken (engl. *standardization*)

(P4) Um die theoretischen RBAC-Referenzmodelle anwendbar zu machen, werden die in ihnen aufgeführten Komponenten in drei Bereiche einsortiert. Die Autoren definieren strukturelle, funktionale sowie informative Komponenten und basieren auf RBAC96. Strukturelle Komponenten sind die statischen Teile des Referenzmodells, namentlich sind das Benutzer, Rollen, Berechtigungen, Rollenhierarchien, Objekte, Operationen und Einschränkungen und beziehen sich einerseits auf deren Semantik, die im Referenzmodell bereits klar definiert ist, aber auch deren Syntax. Funktionale Komponenten stellen Verknüpfungen zwischen den erstgenannten Komponenten her, was namentlich Benutzerzuweisungen und Berechtigungszuweisungen zu Rollen sind. Zusätzlich werden zwei Funktionen eingeführt, die erstens auf Benutzern und zweitens auf Rollen operieren und als Ergebnis die mit ihnen verbundenen strukturellen Objekte zurückliefern. Informationskomponenten spiegeln relationale Datenbanken bzw. Verzeichnisdienste als Datenhaltung für die erstgenannten Komponenten und den Beziehungen zwischen ihnen wieder.

Lösungen

Was sind die eigenen Lösungen?

(L1) RolePartner stellt ein Technologiewerkzeug dar, welches das Definieren von Rollen zusammen mit Hierarchien und Benutzern ermöglicht und orientiert sich damit am erweiterten RBAC-Modell. Es verfolgt eine 3-Tier Architektur bestehend aus UI, Diensten und dem Netzwerkinterface als Schnittstelle zur Datenbankschicht.

(L2) Der eigene Ansatz teilt die Verwaltung in die beiden diskreten Bereiche *role engineering* und *role administration*.

(L3) Ein zentraler Aspekt des RolePartners ist das logische Trennen der Datenhaltung der RBAC-Policy von dem Ort der Anwendung. Durch diese Trennung wird die Interoperabilität gewährleistet.

(L4) Java-basierte Anwendung

Nachweise

Welche Nachweise werden hinsichtlich der Tragfähigkeit der eigenen Lösungen geliefert?

(N1) Der beschrieben Ansatz resultiert in einer Java-basierten Anwendung namens RolePartner mit vier vordefinierten Administratorenrollen und klar voneinander abgegrenzten Befugnissen innerhalb des Rollenmanagements. Es gibt allerdings auch eine Rolle, die Vollzugriff auf das Gesamtsystem hat. Sie wird in ein PMI mit X.509 Zertifikaten implementiert und hinter die Schnittstelle zum Zertifikatsserver für Attribute, der für X.509 benötigt wird, versteckt.

(N2) RolePartner stellt die Hierarchie in den Mittelpunkt und kann mit den Prinzipen der Einschränkungen (engl. *constraints*) und SoD umgehen.

Offene Fragen

Welche Fragen sind noch ungelöst geblieben bzw. stellen sich dem Leser?

(O1) Es bleibt unklar, ob RolePartner alle drei Arten des *role engineering* (*top down*/*bottom up*/*middle out*) unterstützt.

(O2) Wie bringt RolePartner die Sicherheitsadministration und die Systemadministration zusammen, wo einerseits Sicherheits-Policys und andererseits die Verwaltung von Informationssystemen im Vordergrund stehen?

(O3) Wieso genau vier Admin-Rollen?

(O4) Wieso verwendet RolePartner trotz der Aufteilung in vier diskrete Administratorrollen dann zusätzlich noch einen Super-Admin?

(O5) Was genau versteht RolePartner unter dem Begriff „Hierachie“? Ist damit lediglich eine hierarchische Struktur gemeint, oder auch eine explizite Vererbung von Rechten innerhalb der Hierarchie? Können Benutzerrechte vererbt werden?

(O6) Insgesamt bietet RolePartner nur vier Funktionen: Das Verwalten von Benutzer/Rollen-Relationen und Rollen/Berechtigung-Relationen. Existiert die Möglichkeit von Berichten (engl. *reports*)

(O7) RolePartner ist durch seinen Fokus auf eine zentralisierte Verwaltung von Sicherheits-Policys eher für Autorisierungsumgebungen mit starkem Bezug zu Sicherheits-Policys geeignet. Eine Dezentralisierung für große Unternehmen wäre daher wünschenswert, was aber durch RolePartner nur in beschränktem Maße unterstützt wird. Einen möglichen Ansatz hierfür bietet Sandhu in [SB+99]. Auch wäre es im Bezug auf zukünftige RBAC-Komponenten wünschenswert, klar verständliche Einschränkungen anzubieten. Beispielsweise unterstützt RolePartner statisches SoD und kann mit dynamischem SoD nicht umgehen.

(O8) Kann man Begriffe wie „Rolle“ in ein semantisch und syntaktisch so steifes Kostüm zwängen, wie es RolePartner tut? Ist es dann nicht schon zu steif, um in unterschiedlichen Umgebungen individuell angepasst werden zu können?

G Literatur

[Ba96] Helmut Balzert: *Lehrbuch der Software-Technik: Software-Entwicklung*, Spektrum, Akademischer Verlag, 1996

[Ba98] Helmut Balzert: *Lehrbuch der Software-Technik: Software-Management, Software-Qualitätssicherung*, Unternehmensmodellierung, Spektrum, Akademischer Verlag, 1998

[Bun98] Deutscher Bundestag: *Gesetz zur Kontrolle und Transparenz im Unternehmensbereich*, http://www.beckmannundnorda.de/kontrag.html, Webseite, Stand: 25. Mai 2008

[C&M-A-BA] Cooperation & Management: *Advanced Web Applications (AWA) Basics of AWA, Course Uni of the Lecture 'Advances Web Applications'*, http://www.cm-tm.uka.de, Universität Karlsruhe (TH), C&M (Prof. Abeck), 2008.

[C&M-A-ID] Cooperation & Management: *Advanced Web Applications (AWA) Identity management, Course Unit of the Lecture 'Advances Web Applications'*, http://www.cm-tm.uka.de, Universität Karlsruhe (TH), C&M (Prof. Abeck), 2008.

[Di08] Aleksander Dikanski: *Integration eines bestehenden Sicherheitsprodukts in eine bestehende Zugriffskontroll-Architektur*, Diplomarbeit, Forschungsgruppe C&M, Universität Karlsruhe (TH), 2008

[DM08] Aleksander Dikanski, Korbinian Molitorisz: *Autorisierungsprüfung für Webservices mit IBM-Werkzeuge*, Team-Studienarbeit, Forschungsgruppe C&M, Universität Karlsruhe (TH), 2008

[Du06] Dudenredaktion: *Die deutsche Rechtschreibung - Das umfassende Standardwerk auf der Grundlage der neuen amtlichen Regeln*, Band 1, Bibliographisches Institut, Mannheim, 24. Auflage, Juli 2006

[EB+07] Christian Emig, Frank Brandt, Sebastian Abeck, Jürgen Biermann, Heiko Klarl: *An Access Control Metamodel for Web Service-Oriented Architectures*, International Conference on Software Engineering Advances, 2007

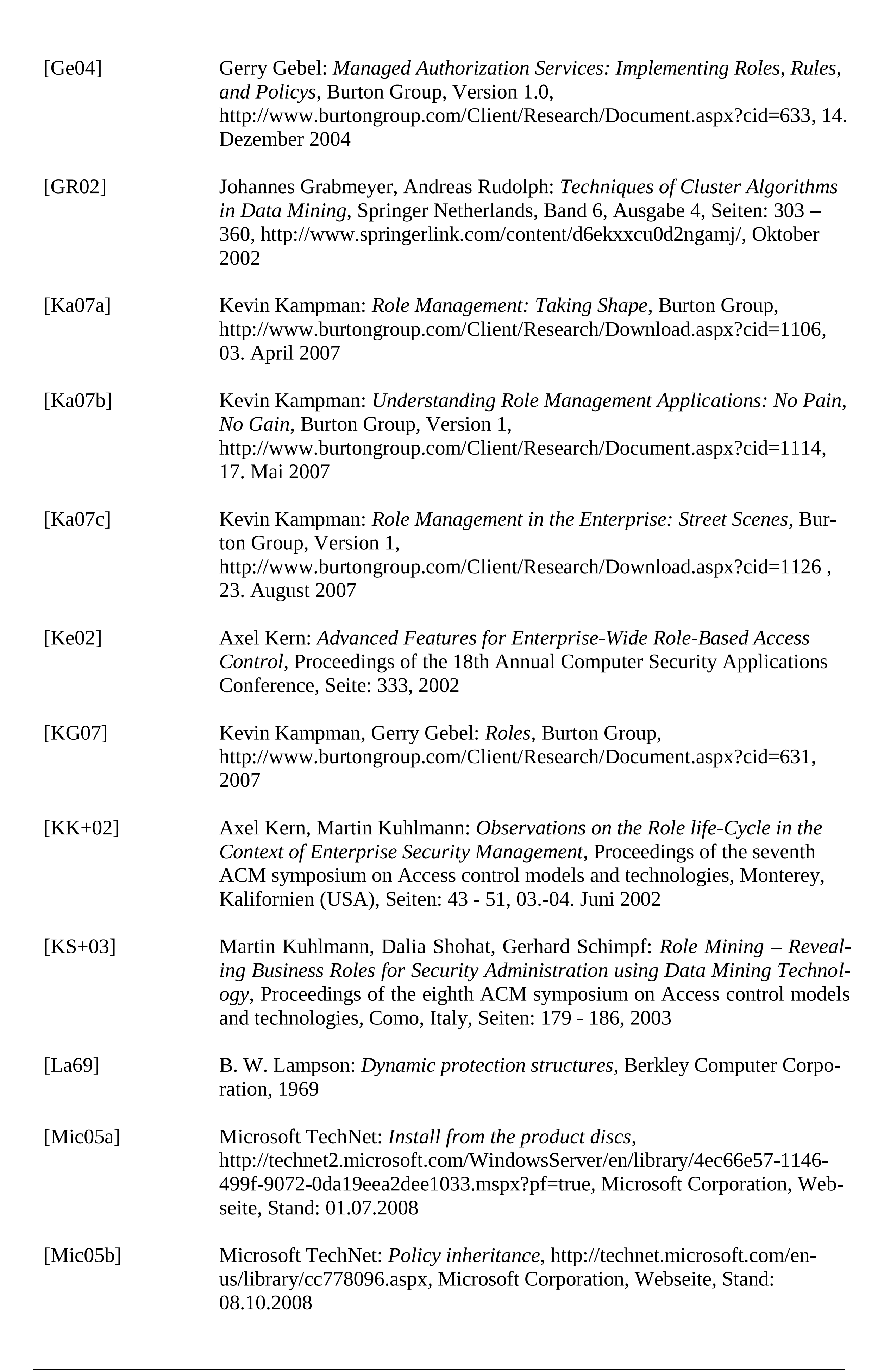
[Ec05] Claudia Eckert: *IT-Sicherheit*, Oldenbourg, 2005

[Em08] Christian Emig: *Zugriffskontrolle in dienstorientierten Architekturen*, Universität Karlsruhe, 2008

[FK+07] David F. Ferraiolo, D. Richard Kuhn, Ramaswamy Chandramouli: *Role-Based Access Control*, *Second Edition*, Artech House, 2007

[FS+01] David F. Ferraiolo, Ravi Sandhu, Serban Gavrila, D. Richard Kuhn, Ramaswamy Chandramouli: *Proposed NIST Standard for Role-Based Access Control*, ACM Transactions on Information and System Security (TISSEC), Band 4 , Ausgabe 3, Seiten: 224 - 274, August 2001

[Ga02] Jay R. Galbraith: *Designing Organisations. An Executive Guide to Stratgy, Structure, and Process*, John Wiley & Sons, Inc., 2002



[Ge04] Gerry Gebel: *Managed Authorization Services: Implementing Roles, Rules, and Policys*, Burton Group, Version 1.0, http://www.burtongroup.com/Client/Research/Document.aspx?cid=633, 14. Dezember 2004

[GR02] Johannes Grabmeyer, Andreas Rudolph: *Techniques of Cluster Algorithms in Data Mining*, Springer Netherlands, Band 6, Ausgabe 4, Seiten: 303 – 360, http://www.springerlink.com/content/d6ekxxcu0d2ngamj/, Oktober 2002

[Ka07a] Kevin Kampman: *Role Management: Taking Shape*, Burton Group, http://www.burtongroup.com/Client/Research/Download.aspx?cid=1106, 03. April 2007

[Ka07b] Kevin Kampman: *Understanding Role Management Applications: No Pain, No Gain*, Burton Group, Version 1, http://www.burtongroup.com/Client/Research/Document.aspx?cid=1114, 17. Mai 2007

[Ka07c] Kevin Kampman: *Role Management in the Enterprise: Street Scenes*, Burton Group, Version 1, http://www.burtongroup.com/Client/Research/Download.aspx?cid=1126 , 23. August 2007

[Ke02] Axel Kern: *Advanced Features for Enterprise-Wide Role-Based Access Control*, Proceedings of the 18th Annual Computer Security Applications Conference, Seite: 333, 2002

[KG07] Kevin Kampman, Gerry Gebel: *Roles*, Burton Group, http://www.burtongroup.com/Client/Research/Document.aspx?cid=631, 2007

[KK+02] Axel Kern, Martin Kuhlmann: *Observations on the Role life-Cycle in the Context of Enterprise Security Management*, Proceedings of the seventh ACM symposium on Access control models and technologies, Monterey, Kalifornien (USA), Seiten: 43 - 51, 03.-04. Juni 2002

[KS+03] Martin Kuhlmann, Dalia Shohat, Gerhard Schimpf: *Role Mining – Revealing Business Roles for Security Administration using Data Mining Technology*, Proceedings of the eighth ACM symposium on Access control models and technologies, Como, Italy, Seiten: 179 - 186, 2003

[La69] B. W. Lampson: *Dynamic protection structures*, Berkley Computer Corporation, 1969

[Mic05a] Microsoft TechNet: *Install from the product discs*, http://technet2.microsoft.com/WindowsServer/en/library/4ec66e57-1146-499f-9072-0da19eea2dee1033.mspx?pf=true, Microsoft Corporation, Webseite, Stand: 01.07.2008

[Mic05b] Microsoft TechNet: *Policy inheritance*, http://technet.microsoft.com/en-us/library/cc778096.aspx, Microsoft Corporation, Webseite, Stand: 08.10.2008

[Mic08] Microsoft Developer Network: Web Part, http://msdn.microsoft.com/en-us/library/bb815365.aspx, Microsoft Corporation, Webseite, Stand: 07.11.2008

[OIM07a] Omada A/S: *Omada Enterprise & Omada Identity Manager 6.0 Installation Guide*, Omada A/S, Version 3, 07.12.2007

[OIM07b] Omada A/S: *Omada Identity Manager Extensible Management Agent*, Omada A/S, Version 2, 12.12.2007

[OIM08] Omada A/S: *Omada Transport System Administrator Guide*, Omada A/S, Version 2, 01.02.2008

[OMG07] Object Management Group: *OMG Unified Modeling Language (OMG UML),Superstructure*, Object Management Group, Version 2.1.2, November 2007

[SA+03] Dongwan Shin, Gail-Joon Ahn: *A Role Administration System in Role-based Authorization Infrastructures – Design and Implementation*, Proceedings of the 2003 ACM symposium on Applied computing, Melbourne, Florida (USA), 2003

[SA+04] Dongwan Shin, Gail-Joon Ahn, Sangrae Cho, Seunghun Jin: *A role-based infrastructure management system: design and implementation*, John Wiley & Sons Ltd, 2004

[SB+99] Ravi S. Sandhu, Venkata Bhamidipati, Quamar Munawer: *The ARBAC97 model for role-based administration of roles*, ACM Transactions on Information and System Security, Kapitel 2(1), Seiten 105-135, 1999

[SB08] Ravi S. Sandhu, Venkata Bhamidipati: The ASCAA Principles for Next-Generation Role-Based Access Control, International Symposium on Collaborative Technologies and Systems 2008, Seite 532, 19. – 23. Mai 2008

[SC+96] Ravi S. Sandhu, Edward J. Coynek, Hal L. Feinsteink, Charles E. Youmank: *Role-Based Access Control Models*, IEEE Computer, Band 29, Number 2, Seiten 38-47, Februar 1996

[Sen02] The Senate and House of Representatives of the United States of America in Congress: *Sarbanes-Oxley Act*, Washington D.C., 23. Januar 2002, http://frwebgate.access.gpo.gov/cgi-bin/getdoc.cgi?dbname=107_cong_bills&docid=f:h3763enr.tst.pdf, Webseite, Stand: 25. Mai 2008

[Sun08a] Sun Microsystems: *Sun Welcomes Vaau*, http://www.sun.com/software/vaau/index.xml, Sun Microsystems, Webseite, Stand: 25. Mai 2008

[Sun08b] Sun Microsystems: *Product line Identity Management*, http://www.sun.com/software/index.jsp?cat=Identity%20Management&tab=3, Sun Microsystems, Webseite, Stand: 29. Mai 2008-06-01

[Ta02] Andrew S. Tanenbaum: *Moderne Betriebssysteme, 2., überarbeitete Auflage*, Pearson Studium, 2003

[Vaau07a] Vaau Incorporated: *RBACx Install Guide 4.0*, Vaau Incorporated, 2007

[Vaau07b] Vaau Incorporated: *RBACx User Guide 4.0*, Vaau Incorporated, 2006

[WW07] Felix Wortmann, Robert Winter: *Vorgehensmodelle für die rollenbasierte Autorisierung in heterogenen Systemlandschaften*, Wirtschaftsinformatik Band 49, Nummer 6 Seiten 439-447, Dezember 2007

Die angegebenen Webseiten wurden am 14. Oktober 2008 auf Gültigkeit überprüft.